U.S. Army Research Institute
for the Behavioral and Social Sciences

Research Report 1543

Organization of Technical Information for Electronic Delivery

John E. Mears, Richard Braby, Joseph Hebert,
and A. J. G. Babu
Eagle Technology, Inc.

J. Peter Kincaid and Larry L. Meliza
U.S. Army Research Institute

October 1989

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

Research accomplished under contract
for the Department of the Army

Eagle Technology, Inc.

Technical review by

Dwight J. Goehring
Bruce W. Knerr

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS --
2a. SECURITY CLASSIFICATION AUTHORITY --	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE --	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) --	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARI Research Report 1543
6a. NAME OF PERFORMING ORGANIZATION Eagle Technology, Inc. 6b. OFFICE SYMBOL (If applicable) --	7a. NAME OF MONITORING ORGANIZATION U.S. Army Research Institute Orlando Field Unit
6c. ADDRESS (City, State, and ZIP Code) 950 North Orlando Avenue Winter Park, FL 32789	7b. ADDRESS (City, State, and ZIP Code) 12350 Research Parkway Orlando, FL 32826-3276
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences 8b. OFFICE SYMBOL (If applicable) PERI-I	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N61339-86-D-0009-0005
8c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 63744A; PROJECT NO. 795; TASK NO. 348; WORK UNIT ACCESSION NO. C1

11. TITLE (Include Security Classification)
Organization of Technical Information for Electronic Delivery

12. PERSONAL AUTHOR(S) Mears, John E.; Braby, Richard; Hebert, Joseph; and Babu, A.J.G. (Eagle Tech., Inc.); Kincaid, J. Peter, and Meliza, Larry L. (ARI)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 87/07 TO 89/01	1989, October	

16. SUPPLEMENTARY NOTATION
Current Contracting Officer's Representative is Larry L. Meliza.

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Technical Information; Procedure Simulation; Job Performance Aids; Visual Authoring System; Microcomputers
(Continued)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The U.S. Army is preparing to make the transition to electronic technical manuals. The objectives of this effort were to design, create, and demonstrate formats for displaying technical information on microcomputer screens. These formats were to take maximum advantage of the display and interaction capabilities that set computers apart from the paper media generally used for this type of information. In addition, the formats were to be economical to produce, store, retrieve, and display.

The resulting formats are in the form of job performance aids and procedure simulations. Job performance aid display algorithms retrieve text and graphic data elements from data files and display the data in different formats for technicians of differing skill levels. Depending upon information needs, the technician can review written descriptions of task steps, call up static graphics and animated graphics to clarify procedural text, call up locator graphics to help locate a particular item of equipment in the context of a larger system, and/or call up detailed descriptions of task steps that are so complex that they
(Continued)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Larry L. Meliza	22b. TELEPHONE (Include Area Code) (407) 380-4374
	22c. OFFICE SYMBOL PERI-IF

DD Form 1473, JUN 86 *Previous editions are obsolete.*

ARI Research Report 1543

18. SUBJECT TERMS (continued)

Hypertext
Object Oriented Programming
Visual Programming

19. ABSTRACT (continued)

warrant being addressed as a task in their own right. Formats for "smart" display frames used in procedure simulations include "locator screens," which can be used to train a technician to identify equipment components, and "action screens," which can be used to train a technician to check/operate equipment.

The job performance aid and procedure simulation formats were demonstrated through development of two rudimentary information delivery and authoring software programs. Army organizations expressed interest in using the software as a research tool. The government developed user's guides for the software, assessed the extent to which the software was user-friendly, and identified issues for future research.

Research Report 1543

Organization of Technical Information for Electronic Delivery

John E. Mears, Richard Braby, Joseph Hebert, and A. J. G. Babu
Eagle Technology, Inc.

J. Peter Kincaid and Larry L. Meliza
U.S. Army Research Institute

Field Unit at Orlando, Florida
Steven L. Goldberg, Chief

Training Research Laboratory
Jack H. Hiller, Director

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

October 1989

Army Project Number
2Q263744A795

Training and Simulation

FOREWORD

The U.S. Army is preparing to make the transition to electronic technical manuals. Defining the formats for providing technical data on microcomputer screens is an important part of this transition process.

This report describes and illustrates screen formats designed to meet specified human factors guidelines. Each screen addresses a small number of procedural steps and allows the user to call up static graphics, action graphics, locator graphics, and detailed procedural steps, as needed.

This project was sponsored by the U.S. Army Project Manager for Training Devices (PM TRADE). The results of this report were briefed to the sponsor and the U.S. Army Armament Research and Development Engineering Center (ARDEC) during the ARDEC review of software considerations for the Militarized/ Miniaturized Electronic Information Delivery System (MEIDS).

EDGAR M. JOHNSON
Technical Director

ORGANIZATION OF TECHNICAL INFORMATION FOR ELECTRONIC DELIVERY

EXECUTIVE SUMMARY

Requirement:

To design, create, and demonstrate formats for displaying procedural technical information on microcomputer screens. These formats were to take maximum advantage of the display and interaction capabilities that set computers apart from the paper media generally used for this type of information. In addition, the formats were to be economical to produce, store, retrieve, and display.

Procedure:

Formats for three types of procedures were developed: equipment operation, proceduralized troubleshooting, and emergency procedures. Researchers applied innovative job performance aid (JPA) formats to the first two types of procedures. JPA formats provide a combined graphic text and graphic presentation of a procedure that the user may browse at any of several levels of detail. The screen displays for these formats were evaluated to determine their compliance with relevant human factors guidelines. Emergency procedures require that the technician perform the procedure quickly and from memory. To support this type of procedure, formats for procedure simulation were developed. Procedure simulation formats present a dynamic graphic model of a piece of equipment, allowing the user to practice the steps of the procedure. Rudimentary authoring systems were developed for both JPA and procedure simulations to support the demonstration of formats.

Army interest in the screen formats resulted in questions about the work required to develop the two rudimentary software packages into user-friendly software and requests for copies of the software. The government developed user's guides addressing use of the delivery systems and authoring systems for each of the contractor-produced software packages, and it provided a critique of the software, including recommendations for further research.

Findings:

Efficient browse and access techniques were developed for JPAs based on database structure instead of storing the data in a traditional frame-by-frame fashion. Display algorithms were developed to retrieve text and graphic data elements from data differing skill levels. Depending upon information needs, the technician can review written descriptions of task steps, call up static graphics and animated graphics to clarify procedural text, call up "locator graphics" to help locate a particular item of equipment in the context of a larger system, and/or call up detailed descriptions of task steps that are so complex that they warrant being addressed as a task in their own right.

For procedure simulations, formats for "smart" display frames were developed that can be used to support proceduralized training on a variety of equipment. These formats include "locator screens," which can be used to train a technician to identify equipment components, and "actions screens," which can be used to train a technician to check/operate equipment.

For both types of formats, user-friendly authoring systems were developed and used to create demonstration formats. These authoring tools were prepared using an object-oriented approach in the Smalltalk/V programming language operating on IBM PC/AT compatible microcomputers. The rudimentary information delivery and authoring software requires refinement to address certain unfriendly menus and disk swapping problems.

Utilization of Findings:

This report may be used by researchers for guidance in formatting technical information for electronic delivery and in the design and development of authoring systems to produce such materials. Specific applications of these findings include the development of electronic technical manuals and low-cost procedural simulations.

ORGANIZATION OF TECHNICAL INFORMATION FOR ELECTRONIC DELIVERY

CONTENTS

CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF FIGURES (Continued)

Page

LIST OF FIGURES (Continued)

Page

LIST OF FIGURES (Continued)

Page

LIST OF FIGURES (Continued)

ORGANIZATION OF TECHNICAL INFORMATION FOR ELECTRONIC DELIVERY

SECTION I: INTRODUCTION

Objective

The objective of this project was to create and demonstrate formats for displaying procedural information on microcomputer screens. The procedural information to be displayed is that required to guide technicians in performing tasks such as troubleshooting and repair of equipment. The potential applications of these electronic displays include replacement of paper-based technical manuals and the conduct of procedure training with simulations.

The display formats must be designed to take advantage of the potential benefits of the electronic medium over the paper-based medium, and they must address human factors principles for screen design. The formats designed must be economical to produce, store, retrieve and display.

This project was sponsored by the U.S. Army Project Manager for Training Devices (PM TRADE) in its role as the initial proponent for the Militarized/Miniaturized Electronic Information Delivery System (MEIDS). The MEIDS is intended to be a hand-held computer for use in provided technical information and training in the field environment. However, the findings of this project are also relevant to the Electronic Information Delivery System (EIDS) designed for use in the laboratory and shop environments.

Background

DOD Emphasis on Computer Delivered Technical Information. The U.S. Department of Defense is committed to replacing paper-based delivery of technical information (TI) with electronic delivery. The goal of current policy is to make significant progress towards this goal by 1990, as articulated in a memorandum regarding the Computer-Aided Logistic Support (CALS) program by Under Secretary of Defense William H. Taft IV, dated 24 September 1985.

> "The DOD-Industry Task Force on Computer-aided Logistic Support has recommended action....to improve the accuracy, timeliness, and use of logistic technical information. ...I have approved a strategy for transitioning from our current paper-intensive weapon system support process to a largely automated and integrated mode of operation with substantial progress by the end of this decade. It is my

goal that the Department of Defense will establish plans to acquire, process, and use logistic technical information in digital form. Insofar as possible, this shall be accomplished for new weapon systems entering production in 1990 and beyond"

One major goal of CALS is to accelerate the automation of contractor processes for generating logistic technical information products (technical manuals and training materials). A second major goal is to rapidly increase U.S. military capabilities to receive, distribute, and use logistic technical information in digital form to improve weapon system maintenance and training. Much development needs to occur within the next few years to accomplish these goals.

Army, Navy and Air Force Projects. The Army, Navy and Air Force are involved in projects to field systems which will deliver electronically the types of technical information found in technical manuals, job performance aids, and certain training materials. In general, these efforts concern development of delivery devices, creating authoring systems, loading sample data bases with technical information, and conducting field trials of the system.

One project, sponsored by the Joint Service Manpower Research and Development Program and executed by the Naval Training Systems Center and the Army Research Institute, was to develop and field test one system for the electronic delivery of technical information, the Personal Electronic Aid for Maintenance (PEAM). The PEAM is a proof-of-concept type portable microcomputer that presents training and aiding information through graphic displays and synthesized speech, including software support for the database.

Schurman and Kincaid (1987) evaluated the PEAM job aid delivery system for the Army. In the evaluation, the PEAM was used by technicians working on the M1 Abrams Tank Fire Control System. The conclusions reached in this evaluation included the finding that this form of information delivery is usable and reduced errors 5:1 for troubleshooting tasks and 2:1 for common corrective maintenance tasks when compared to maintenance being performed with traditional paper technical manuals. Electronic delivery proved to be highly acceptable to maintenance technicians.

The development of a delivery device is the goal of the Army's (MEIDS) project. The Army plans that MEIDS will be the portable electronic device which will largely replace the technical manuals that Army technicians use in the field. The configuration for the device is still being formulated, but it will be microcomputer-based, rugged and portable, use an expert system for indexing and accessing data stored in its databases, and contain many illustrated job aids.

The Navy recently conducted field tests of a computer-aided job aid called Fault Isolation by Nodal Dependency (FIND), which is a part of the Naval Technical Information Presentation Program (NTIPP). Fuller, Post, and Mavor (1987) made two observations. First, ninety percent of the subjects preferred electronically delivered TI over paper-based TI. Second, FIND allowed more accurate and, in certain cases, faster performance than conventional paper-based TI. In a separate study, Fuller, LeBeau, Mavor, Post and Sawyer (1987) found that 92 % of the subjects preferred electronic TI, and the subjects were able to perform more accurately and at a higher speed.

As part of the NTIPP development effort, Thomas, Braby and Mears (1988) developed a microcomputer-based method of practicing emergency procedures. Emergency procedures, casualty procedures and other similar procedures form a class of operations that must be performed rapidly and accurately from memory or with the aid of a simple checklist. Two types of technical information support this class of human performance. First, the procedure can be supported with job aids appropriate for equipment operation. Second, procedure simulations can be provided so that the technicians can practice the procedures in order to commit them to memory in such a way that the procedure can be recalled under stressful conditions and performed when required. The study by Thomas, Braby and Mears further documents the generally accepted belief that the use of microcomputer type procedure simulation is one way to provide practice opportunities, either independent of other practice opportunities or as a pretraining step before practice on complex simulators or on the operational equipment. While the study concerned both types of TI, of interest here is the procedure simulation. Design principles were postulated, two procedure simulations were created according to the design principles, and field trials in the form of summative evaluations were conducted. The results indicated that the formats used in creating the procedure simulations were effective and that both instructors and students considered the simulations to be highly useful.

The Air Force is also developing a system to author and display job performance aid technical information via portable computers. This AFHRL project is called the Authoring and Presentation System (APS). A team of programmers is working on-site at Wright-Patterson Air Force Base to create this system. One of the important characteristics of APS is that it incorporates a relational database approach allowing the creation of individual display frames from databases rather than requiring the authoring and storage of a massive number of individual frames.

Lessons Learned from Experience with Paper-based Fully Proceduralized Job Aids. The armed forces' experience with paper-based job performance aids also provides useful background information for designing computer-delivered job performance aids. Substantial research has been conducted in this area. One basic type of job aid is the Fully Proceduralized Job Performance Aid (FPJPA). With this type of job aid, an apprentice technician is presented with highly illustrated information on how to perform a step in the procedure. The apprentice performs that step, referring back to the information as needed. In a step-by-step manner, the apprentice performs the entire procedure.

When FPJPAs are extensively used within a program, certain problems emerge. Technicians can perform the proceduralized tasks, but they tend to feel that using these detailed job aids does not enhance their career skills. Wider use of FPJPAs has not taken place, in part, because of this user perception.

Post and Price (1973) list three characteristics of a good work environment; characteristics that appear to be particularly relevant for user acceptance of FPJPA or any other form of job performance aid. These characteristics are listed and described below.

(1) Opportunity to Learn. A good job (job aid) is one which affords opportunities to acquire skills or knowledge relevant to career advancement.

(2) Challenge. Workers tend to be satisfied with jobs (job aids) that, over a period of time, offer challenge. The challenge can occur in the form of quantity or complexity of tasks included in the job.

(3) Meaningfulness of the Work. Workers tend to feel satisfied when they have a significant role in performing tasks that they perceive to be critical to the output or goal of their organization.

The typical use of FPJAs fails to meet the above criteria. Chernoff, Joyce and Nauta (1985), in a report on the status and future directions of maintenance job aids, discuss this issue.

"Perhaps the most important factor that was ignored in the early JPA efforts was the needs of the maintenance workers themselves. The 'technicians" felt degraded since they were only qualified to perform the simplest tasks. They felt that they were being 'led by the hand' through a series of steps that had no meaning for them. If a technician did not have these feelings spontaneously, his superiors were quick to show how much more they knew about the system being maintained. He had low status: his abilities were not valued highly; his work had little meaningfulness; instead of deciding what to do on his own, he was

being told at each step what to do. Of course, some needs were being met for some individuals. Some lower mental category individuals were challenged by the job, were meeting with success experiences, and felt that they were making significant contributions in maintaining sophisticated equipment. But for the most part, the experience was demotivating."

Post and Price proposed solutions to these problems in the form of a new type of JPA format. This new format includes both the steps to be performed and information about these steps. It is an attempt to make the work meaningful, and to support the technician with career-relevant knowledge about the equipment and principles governing the performance of the equipment.

In a different study, Nauta (1985) investigated the way the Navy conducts maintenance, and identified problems with Navy maintenance. He also studied support documents, including maintenance JPAs and training, and made recommendation on how to improve the technical information supporting maintenance. Concerning JPAs, he states:

"The most important objective of an aid should be to expedite skill development on the job through emphasizing learning while using the aid. The art of designing an effective aid lies in providing the apprentice-technician sufficient challenge and learning from task performance so that a level of proficiency is reached whereby the aid is no longer needed, except for reference data.."

Nauta (1985) also stated that:

"From a maintenance performance point of view, the sole purpose of an aid should be to wean the technician away from needing the aid....All further R&D in job aiding must be focused on facilitating skill development on-the-job."

<u>The Need to Develop Improved Formats for Electronic Display of Technical Information (TI).</u> At this time, better information is needed on how to format TI for electronic display. An understanding of the way that TI should be organized and presented to the soldier using a new electronic medium is an important prerequisite to the continuing development of computer hardware and software for the electronic delivery of TI.

The research on FPJAs cited above has identified problems in FPJA formats which need to be addressed to ensure that these formats will effectively address the career development concerns of technicians. For example, the aid must be designed to help technicians acquire skills which will" wean them away from needing job aids.

Formats used with paper TI will not work on the computer screen. From a human factors point of view, the two media are significantly different. To use paper-based formats on the computer screen is to set aside considerable research on how to optimize screen design. Using paper-based formats in lieu of formats designed for computer screens will make it difficult for technicians to locate and read required information. Computer-based techniques for data searches and for adapting the display to an individual's needs are forfeited if paper formats are followed.

The cost of producing computer-based technical information is also a major concern. The formats and authoring tools used can make a significant difference in the acquisition and life-cycle cost of fielding the technique. For instance, the high cost of producing paper-based fully proceduralized job performance aids that make extensive use of graphics has kept this form of TI from being more widely used. This high cost is due to at least two factors: (1) an increase in the number of pages required to support a given task, and (2) the more extensive use of expensive graphics. The general expectation is that computer-based job aids will be less expensive. However, computer-based job performance aids could prove to be even more costly to produce than paper job performance aids. This will be true if the job aids are produced frame by frame. One estimate is that it will require 15 to 20 times more electronic frames than paper pages for a given task. While each electronic frame contains less information than a paper page, there would still be a substantial increase in effort to produce electronic frames in the place of paper pages, if the electronic information is stored in frames. To be economical, an authoring system must be built to use data from which individual frames can be constructed, and it must use compression techniques to minimize digital storage requirements.

Organization of Report

In addition to Section I, three other major sections are presented. Section II describes the technical approach used in creating formats for presenting technical information on computer displays. The basic design issues to be resolved are described and the methodologies to be employed are presented.

Section III provides the results of the format design effort relating to job performance aids. This section includes information concerning the types of tasks to be supported, and formats designed for these types of tasks. In addition, the computer algorithms used to implement the formats are described, along with notes on how to use the authoring systems. The latter systems were created to enable non-programmers to prepare technical information for computer delivery according to the described formats. Sample material prepared using the authoring systems is also presented. Section IV provides the results of

the format design effort for procedure simulations. This section contains information concerning procedure simulations similar to that found in Section III. Finally, conclusions and recommendations for the application and further development of the formats and authoring systems are presented in Section V.

The appendixes contain additional material describing the various issues involved with this project. Specifically, Appendix A provides flow diagrams for initial and advanced practice modes of a procedure simulation; Appendix B provides an overview of the Procedure Simulation Produced by Rehearsal (PROSPR) authoring system; and Appendix C provides the human factor guidelines for evaluating the screen formats and an analysis of the JPA formats based on the guidelines.

The appendixes also provide user's guides and critiques of the two demonstration software packages developed during this project. Appendix D addresses the Procedure Job Aid Production System (PJAPS) demonstration software, and Appendix E addresses the PROSPR demonstration software. Each of these appendices covers both the information delivery and authoring systems of the appropriate software.

Page 8 Blank

SECTION II:
TECHNICAL APPROACH

The process used in designing the computer screen formats and display algorithms is delineated in this section. Major steps in this process include the selection of procedures to be demonstrated, the design of formats, the selection of programming techniques and the creation of digital TI.

Although these steps are presented as a linear process, an iterative sequence was followed. Specifically, step 2 through 4 served as a loop in which the experience of creating demonstration materials caused the team members to become aware of new requirements, causing a change in the formats and programming effort.

Choose Procedures to Be Demonstrated

This project focuses on technical information in the form of procedures. Classes of procedures that were considered included equipment operation, repair (i.e., remove and install), scheduled maintenance, system check-out, proceduralized troubleshooting, and emergency procedures. Technical manual and digital type TI for each of these type of procedures were inspected. Each type of procedural TI was a realistic candidate for computer display formats. For this project, three types of procedures were needed that would generate the broadest range of requirements for computer displays. The three classes of procedures selected were equipment operation, proceduralized troubleshooting, and emergency procedures.

A wide range of specific procedures were available to choose from for each of the three classes of procedures. Specific examples of the three types of procedures were selected for the gunner in an M1 tank. system. The specific procedure selected to be demonstrated for each of the three classes of procedures are: "Computer Self Test Performed as a Part of Equipment Turn-on Procedures" as an example of an Equipment Operation Procedure; "Operator Troubleshooting Following Computer Self Test" as an example of a Proceduralized Troubleshooting Procedure; "Computer self test performed under combat conditions" as an "Emergency Procedure"

Digital TI was developed on these procedures as described in the M1 Tank Operator's Manual. This manual is composed of three volumes: Volume 1, Operator Controls and PMCS (TM9-2350-255-10-1); Volume 2, Operation Under Usual and Unusual Conditions (TM 9-2350-255-10-2); Volume 3, Troubleshooting and Maintenance (TM 9-2350-255-10-3). The description of the self test procedures was found in Volume 2, while the troubleshooting procedure was described in Volume 3.

Design Screen Formats

Design guidelines from various sources were considered for use in creating the formats. Certain guidelines came from recently completed field trials of electronically delivered procedural job aids and training materials. Other guidelines came form learning theory and general human factors insights into how to optimize computer displays. Certain of the guidelines apply primarily to job aid design. Still others concern the merging of job aids and training aids into a hybrid medium (i.e., a job aid that trains). Guidelines for two separate sets of formats were required: one for equipment operation and troubleshooting, and one for emergency procedures. Out of this general consideration of design variables, the following high level guidelines were developed for job performance aids and procedure simulations.

Guidelines for the Design of Job Aids. During the course of the project, the screen designs and formats were reviewed at two separate times to ensure compliance with the general human factors guidelines provided in Appendix C. In addition, guidelines specific to frames for electronic job aids were extracted from lessons learned and used to guide the current project. These specific guidelines are listed below.

- Provide for the individual technician to adapt the display of the procedural steps to the level of detail needed to perform the task.
- Display a single step, or no more than a small set of related tasks on the screen at one time.
- Design a system of displays that supports the technician in learning to perform skills (i.e., moving from a need for detailed types of procedural information to being able to perform the procedure with a minimum of reference to TI).
- Provide detailed graphics of the procedural steps for those technicians needing this type of information.
- Provide for an efficient browse capability to allow the technician to easily move through the data without getting lost
- Provide for immediate and easy access to all the step sequences that make up a complete procedure, including those steps common to many procedures.
- Apply human factors principles for efficient screen design

- For reasons of economy, design formats for screens that can be built "on the fly" with data elements pulled from a database rather than storing each frame explicitly.

Guidelines for Designing Procedure Simulation Formats. A completely different set of formats is required for procedure simulations. The emergency procedures type of task to be supported calls for materials to be used by a student during practice. Skilled performance under stressful conditions calls for overlearning of the procedure. This is accomplished through repetitive drill.

Experience with NTIPP emergency procedure simulations during the summative evaluation indicates that two separate modes of instruction are usually required; initial practice and advanced practice. Initial practice is designed for the student with little or no experience with the procedure and helps the student to explore all facets of the procedure and practice segments. Advanced practice is designed for students have learned to perform the operation called for in the initial practice exercises. In advanced practice mode the student is allowed to perform the procedure from start to finish without interruption and with randomly selected faults. Table 1, from Thomas, Braby and Mears (1988), shows some of the similarities and differences between these two modes.

In initial practice, students are allowed to explore how to perform each of the subroutines in the procedure. The student chooses what to practice, and can repeat a subroutine until satisfied. System failures are made to happen frequently. Although this is unrealistic, it allows the student to practice all aspects of the procedure. In addition, the student is encouraged to use any TI normally used on the job. Feedback is provided after every student action. The student must perform an action correctly before being allowed to attempt the next action. When a student makes an error, the system immediately indicates the correct action, and then cycles back requiring the student to properly perform the action. No records are made of errors during initial practice in order to encourage students to explore all aspects of the procedure without fear of recorded errors.

In advanced practice, students perform the procedure as it would be performed on the job. Conditions of the system are established randomly and the student then performs the steps of the procedure in the manner prescribed by the technical manual. The student is encouraged to use any documentation normally used on the job. Feedback is provided after every action; however, the system waits for three successive incorrect responses before prompting the student with the incorrect response. At the conclusion of a pass through a procedure, the system should display the number of errors made.

Procedure simulation flowcharts for initial and advanced practice are provided in Appendix A. These flowcharts record the flow of events used in NTIPP procedure simulations which performed successfully in the carefully observed field trials. They are as presented in Thomas, Braby and Mears (1988) and form the core of the guidelines used in designing the procedure simulation formats and author aids presented in this project.

Table 1.
Characteristics of Initial and Advanced Practice Mode of Procedure Simulation

DIMENSION	Initial Practice	Advanced Practice
1. TRAINING TASK	Explore how to perform a procedure under various conditions.	Perform a procedure, responding to any special conditions that may exist.
2. SCOPE OF EXERCISES	One Procedure.	One Procedure.
3. NUMBER OF RANDOM FAILURES IN PROBLEM	Many	Similar to real equipment.
4. FLEXIBILITY IN CHOOSING STEP SEQUENCE	Choose an step, repeat any step, as desired.	Perform all steps in sequence dictated by technical manual.
5. ROLE OF TECHNICAL MANUAL	Used as job aid while performing procedure.	Used as job aid while performing procedure
6. FEEDBACK	Feedback after every action. Prompts after every error.	Feedback after every action. Prompts after three successive errors.
7. RECORD OF STUDENT PERFORMANCE	No record produced.	Report of number and location of errors during practice run.

Program

Criteria for Prototyping Language. Programming for this project required a prototyping language meeting the following criteria:

- The language's external view should be highly graphical in nature, making it possible to mimic the various control panels, lights, switches, gages, etc., that are a part of the actual equipment.
- A simple, modifiable user interface must be present to ensure use by non-computer specialists and students.
- The language must possess or allow the creation of graphical editors, so that new instances of switches, lights, etc. could be easily created by the author.
- The language must run on today's microcomputers, yet be powerful enough to handle complex situational problems.
- A desirable feature of the prospective language would be incremental compilation: the ability to install small blocks of code without having to compile and link the code for the entire application.

Features of the Smalltalk/V Language Environment. The language selected was Smalltalk/V. Smalltalk/V is a computer language environment complete with compilers, interpreters, graphic editors and text editors.

Smalltalk was first developed at the Xerox Palo Alto Research Center (PARC) in the early 1970's. Refinements produced the most recent version, Smalltalk-80, in 1980. Xerox PARC's main efforts were devoted to making Smalltalk a productive programming environment, while improving user interfaces such as the mouse, and to make the use of graphics, icons, and menus more convenient. A number of "user friendly" computer systems such as the Apple Macintosh can trace their development directly to this programming work.

Smalltalk/V is a language which utilizes many of the concepts of Smalltalk programming, but is able to run on an IBM-PC, PC/XT, PC/AT, or compatibles. The availability within the military of the Zenith 248 microcomputers (PC/AT compatible) and a low cost computer language facilitated the development of the PCGAS.

The Smalltalk/V environment is "object-oriented", which means that all the basic elements of programming (e.g., numbers, strings, data structures, control structures, and procedures themselves) are treated as objects. Objects interact with each other by sending messages.

Smalltalk/V is organized in a class hierarchy structure, where a class is like a factory capable of creating one or more similar objects ("instances"). Each class has an immediate superclass (or parent class) and may have subclasses (children), with the class object at the pinnacle of the hierarchy. Classes higher in the hierarchy are more general in nature, while those located lower in the hierarchy represent more specific structures. For example, "switches" might be at the top of a hierarchy and specific types of switches (e.g., a three-position switch) would be lower in the hierarchy.

The class contains information about the messages that can be performed ("methods") by an instance, and each instance of a class contains internal information which distinguishes it from other instances in that class and maintains information important to the instance ("instance variables"). For example, the class of "three position switches" might take the form of a number of instances, such as differing in terms of appearance. An instance of a class "inherits" all the characteristics (instance variables and methods) of its superclass (e.g., a three position switch or a four position switch has all the characteristics of a switch). By creating a subclass and adding new methods and variables to that subclass, the new class has all of the abilities of its superclass parent(s), but can be instructed to respond to messages specific to itself. Since, objects interact with other objects through messages, one action (e.g., throwing a switch) can trigger several reactions (light comes on, gage moves) from other objects. as one object sends messages to other objects to elicit the responses.

Smalltalk/V also incorporates a built-in graphics editor which makes graphics easy to generate and change. This visual (graphic) aspect is critical to learning procedures since the system must visually depict the various components, interrelationships, and conditions of the equipment during training.

The Smalltalk/V environment is also characterized by overlaid windows, mouse interaction, and pull-down windows. Windows are like rectangular sheets of paper that perform some function, and they can be placed on top of each other on the screen. Choosing a particular menu item from a window causes another window to appear on the screen. Partially obscured windows can be brought to the forefront by positioning the cursor on the window and pressing the appropriate key on the mouse. Smalltalk/V also allows a programmer to view, change or copy most of the methods used in Smalltalk/V's own implementation. For example, if the programmer needed a menu for his applications, he or she could examine the method that Smalltalk/V uses for its own menus, copy the code, and paste the code in his or her application. This allows a programmer to learn Smalltalk/V by example and enhances programming capabilities. Smalltalk/V also allows a programmer to utilize its existing window/mouse

interaction menu format in all application development. Since Smalltalk/V incorporates a mouse-driven interface, a two button mouse is automatically integrated into the system.

Commercial Hardware and Software Used in Project. The software selected and used in this project includes: Smalltalk/V Programming Environment with #1 Application Extensions and EGA/VGA Color Extensions; Eyestar Plus Graphics Scanning Software; ReadRight Optical Character Recognition (OCR) Software. The computer hardware selected and used in the project includes the following:

- Zenith 248 microcomputer (80286 based) with 640K main memory, 360k floppy disk drive, and 20MB hard disk drive.
- Zenith EGA Color Monitor
- Vega Deluxe EGA graphics board with 256k memory.
- Mouse Systems 3-button serial mouse
- Microtek MS-300C Image Scanner
- Hewlett-Packard LaserJet Series II Laser Printer

Demonstrate Formats

Three demonstration programs were created to illustrate the formats created in this project, as listed below.

- Program 1: Job Performance Aid for Equipment Operation Computer Self Test.
- Program 2: Job Performance Aid for Proceduralized Trouble Shooting.
- Program 3: Procedure Simulation of the Computer Self Test as an example of an emergency procedures job aid.

In carrying out these tasks, the necessary sets of formats for dynamic screens were designed. In addition, authoring aids and delivery software were also created. The next two sections describe the design of the formats, the authoring systems created to support the use of the formats, and the delivery of this digital TI.

Programs 1 and 2 were included in a demonstration software package titled "Procedure Job Aid Production System (PJAPS)", and Program 3 was included in a demonstration software package titled "Procedural Simulations Produced by Rehearsal (PROSPR)".

Page 16 Blank

SECTION III:
FORMATS FOR JOB PERFORMANCE AIDS

Job Performance Aid (JPA) screens provide a combined text and graphic presentation of a procedure. Users can browse through the information at any of three levels of detail. A user friendly interface makes it easy to find areas of interest, store relevant steps for later review, or mark a place for quick return after browsing.

This section describes the general characteristics of these screens, as well as the specific characteristics of screens at each of the three levels of detail. After the description, sample screens will be presented.

Format Functions

Unless otherwise noted, the following functions are available on all JPA screens

- Text and Graphics. Nearly every step is accompanied by a line art drawing of the panel or equipment used in performing that step. Drawings are provided on Level 2 and Level 3 screens only.

- Locator Graphics. By simply positioning the cursor on a graphic, and pressing the mouse button, the user can see a drawing of the area around the graphic. This helps the user become oriented to the overall layout of the equipment.

- Animated Graphics. Some Level 3 screens have been animated to allow the user to see how the equipment responds when a step is performed.

- Browsing Controls. The user can easily move forward and backward through the information, or change level of detail.

- Help. The user can access an online help facility which explains the use of the various screen controls.

- Store, Display and Discard. The user can store the current screen image for later recall. The display portion of this function allows the user to page through the stored screens, while the discard function allows individual screens to be dropped from the stored images.

- Mark and Return. By selecting the mark option, the user can place an electronic "bookmark" at the currently displayed screen. By selecting return, the user can go directly to the marked screen from anywhere in the procedure.

- Link to Common Subprocedures. Many procedures are actually performed as part of numerous other procedures. The task of powering up a gunner's station, for example, is performed as the first step in many procedures performed by an M1 tank gunner. Very often such subprocedures are complex enough to warrant being addressed as a procedure in their own right. The system provides links which make these procedures easily available to the user in the context of the main procedure. That is, the system can pull the entire subprocedure into the procedure which the user is currently reviewing, at the user's discretion. Figures 1-1 and 1-2 show the numerous subprocedures involved in the procedures employed in developing sample formats for the current project.

- Procedure Map. This provides an overview of the current procedure, showing where information is available at the various levels and where links to common subprocedures exist.

Screen Formats

Format for Level 1 Screens. The first level of information, intended for the experienced technician, provides the least level of detail while allowing the user to quickly scan the steps of a procedure. No graphics are displayed at this level. One or several steps may appear on each screen, depending on the length of the statements, but only about ten lines of text are presented on any one screen. Except for unusual cases, all the text for a given step appears on a single screen. The user can select any one of the currently displayed steps and proceed to the second level screen for more detailed information on that step. Figure 2 is an example of a Level 1 screen.

Format for Level 2 Screens. The second level of information provides more detail by including graphics with the text. At this level, each screen contains the text for a single step. A line art drawing is displayed showing the panel or other equipment used to perform the step. Arrows are used to indicate points of interest (e.g., buttons to be pressed, lights which come on). Figure 3 is an example of a Level 2 screen.

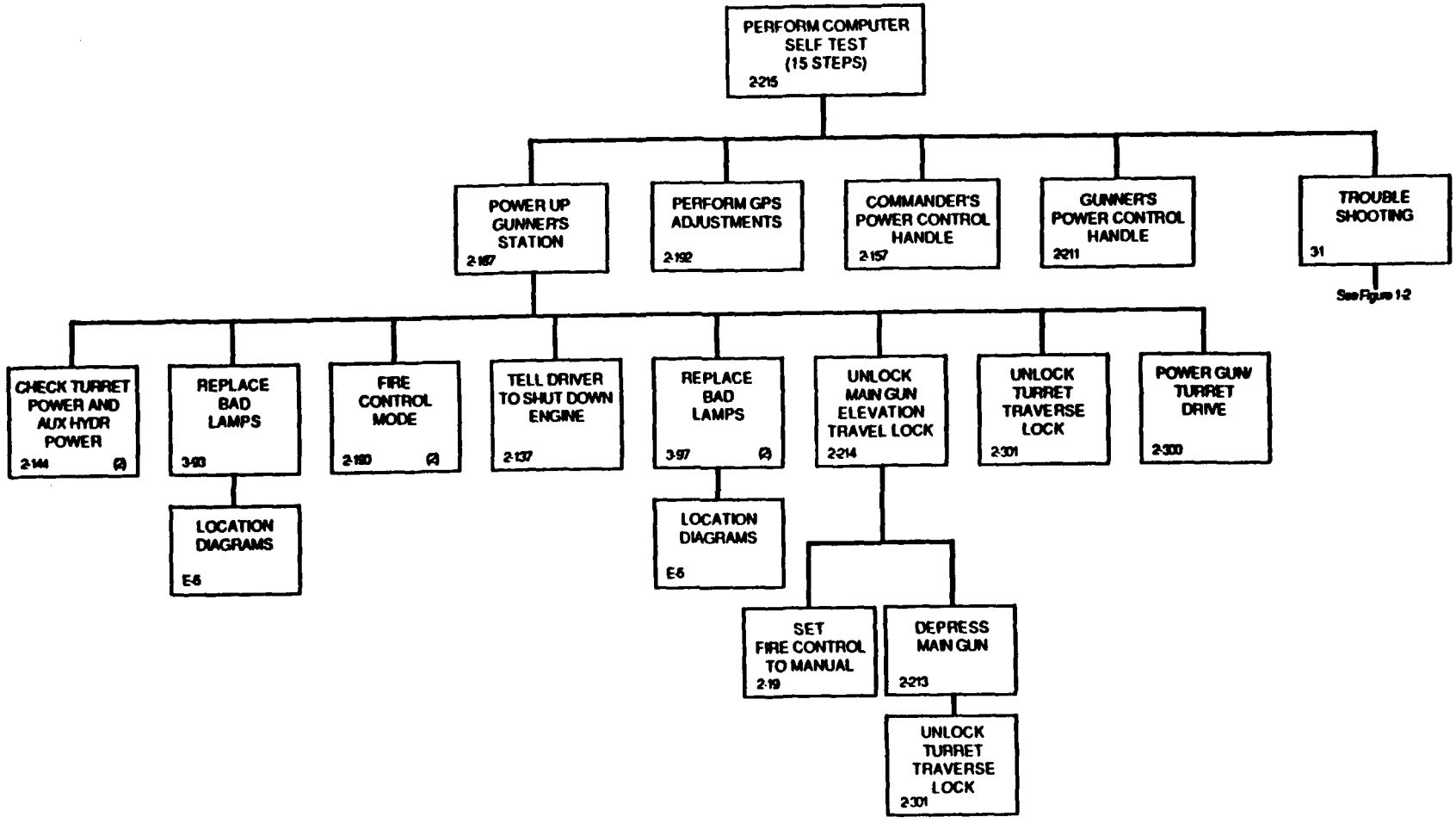


FIGURE 1-1. References To Common Subprocedures In Computer Self-Test Procedure

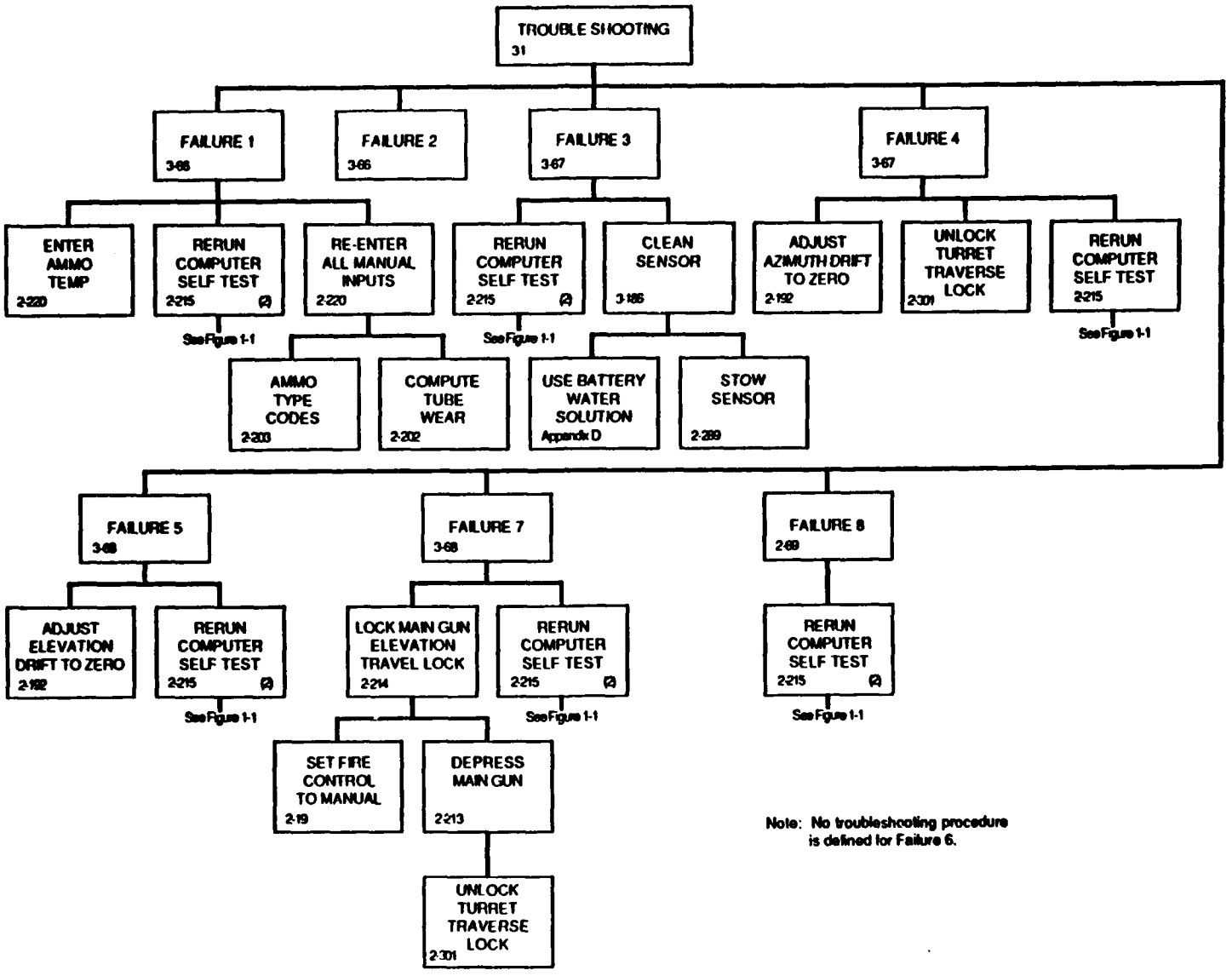


Figure 1-2. References To Common Subprocedures In Troubleshooting Procedures For Computer Self Test

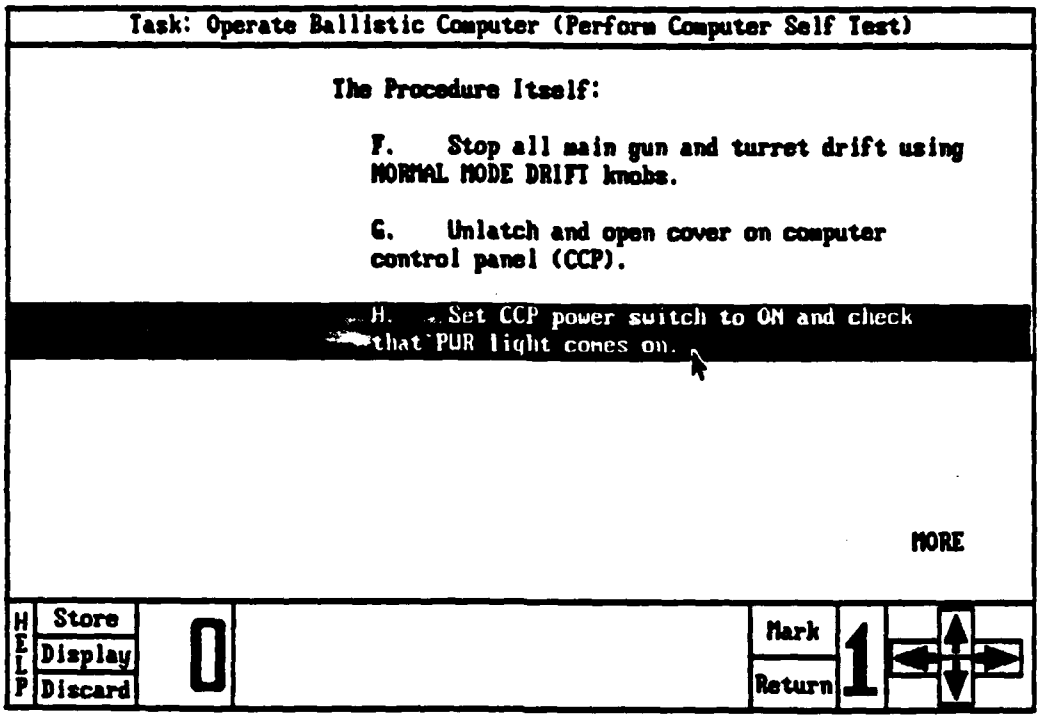


Figure 2. Sample Level 1 Screen

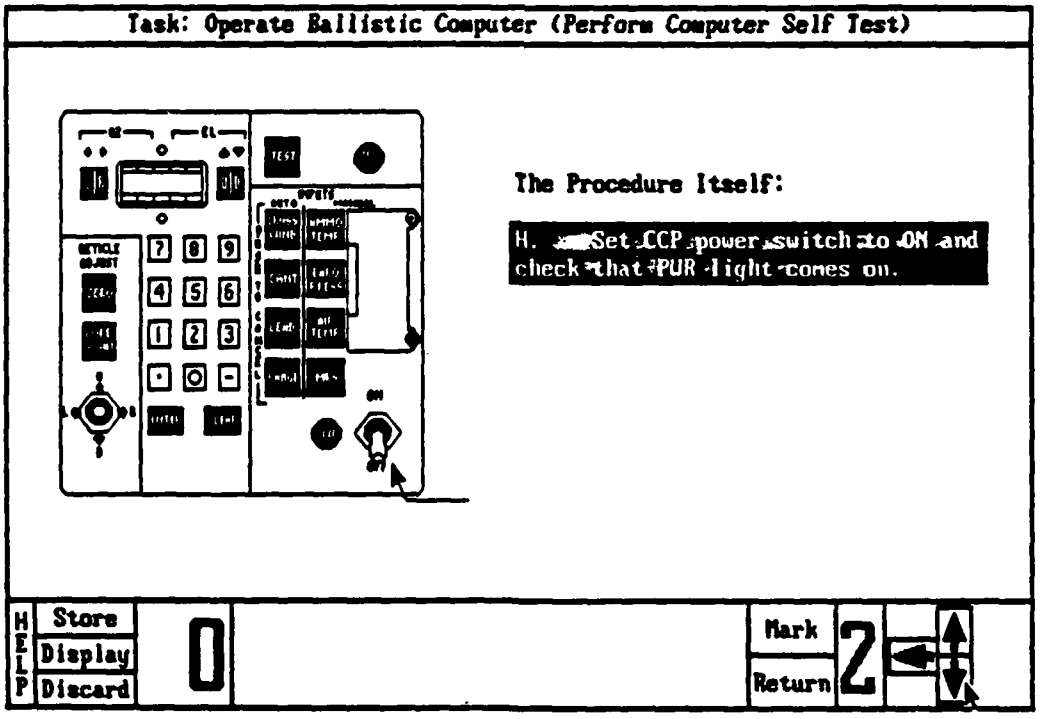


Figure 3. Sample Level 2 Screen with Static Graphics

Format for Level 3 Screens. Third level screens may help to further clarify a procedural step by providing simplified language and/or animated graphics. Animated graphics allow the student to see the actions described in a procedural step by showing what the item of equipment looks like "before" and "after" the action described in the step. These graphics are shown in rapid succession to give the impression of movement. After the student has viewed the action, he or she can reset the screen to the "before" graphic. Figure 4 is an example of a Level 3 screen.

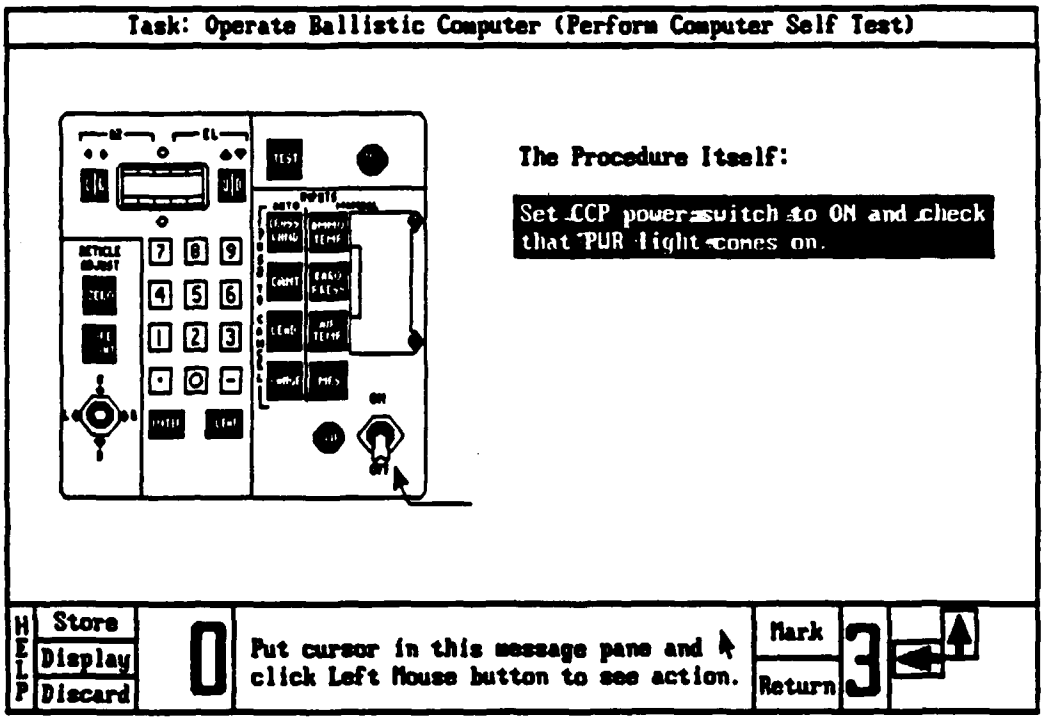


Figure 4. Sample Level 3 Screen with Animated Graphics

Demonstration of Equipment Operation JPA

A brief demonstration of an equipment operation type of JPA is provided in Figures 5-1 through 5-12. For a more comprehensive demonstration of this material, see the User's Guide for the PJAPS Demonstration Software. This guide is provided in Appendix D.

Figure 5-1 shows a level 1 display of the first steps of the procedure "Perform Computer Self Test". The step, "A. Power up gunner's station" is shown in reverse video to indicate that a student has requested more detailed information on this step. A student would "request" additional information by putting the cursor on the text and clicking the left mouse button.

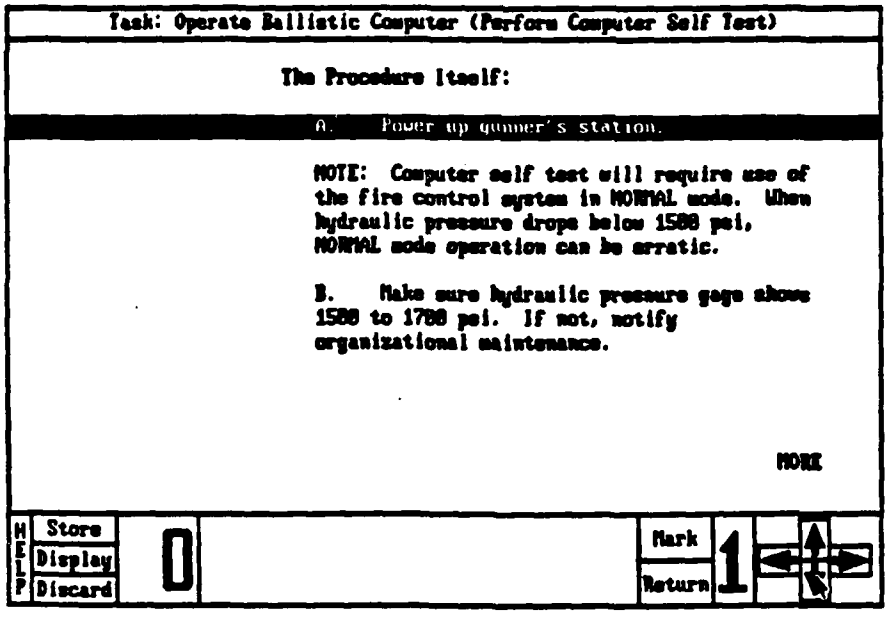


Figure 5-1. Demonstration of Equipment Operation JPA, Screen 1.

"Requesting" information about a procedural step from a level 1 screen may lead to two different results. In certain cases a Level 2 screen will be provided showing graphics. In other cases, where the procedural step is complex enough to be addressed as a procedure in its own right, an entire procedure will be called up. In the present example, "Power up gunner's station" is a complex step which warrants being addressed as a procedure in its own right. Therefore, the new screen called up is essentially a menu for this procedure (Figure 5-2). By selecting the menu item "the procedure itself" and moving the cursor to the down arrow at the bottom of the screen, a student would call up the screen shown in Figure 5-3.

Figure 5-3 displays the first three steps in the procedure "power up gunner's station". It is a Level 1 type display providing a terse checklist of the procedural steps. The word "MORE" in the lower right hand corner of the screen indicates there are more procedural steps. Step A.1 has been selected. In order to display more detailed information on this step the cursor must be moved to the down arrow in the information panel at the bottom of the screen and pressing the left mouse button.

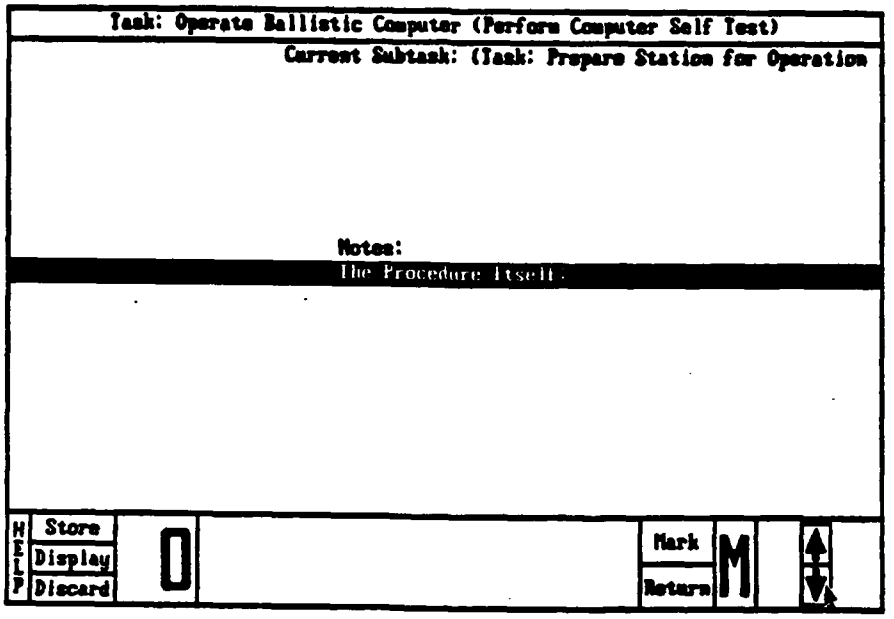


Figure 5-2. Demonstration of Equipment Operation JPA, Screen 2.

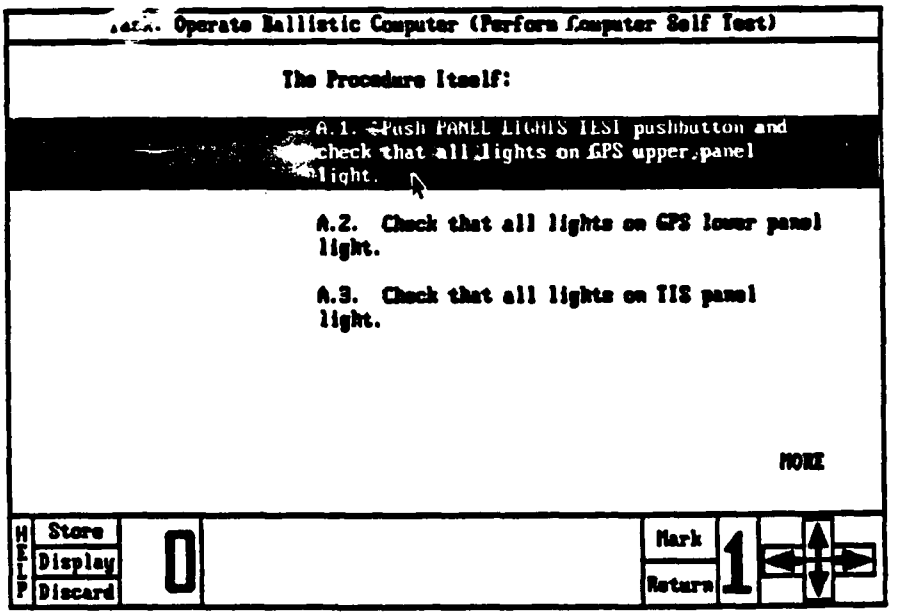


Figure 5-3. Demonstration of Equipment Operation JPA, Screen 3.

Figure 5-4 shows a Level 2 screen providing graphics to help explain the procedural step selected in the previous screen. If a user is uncertain of the location of an assembly, a locator graphic can be called up by placing the cursor on the figure and pressing the left mouse button (see Figure 5-5). If a larger locator view is needed to orient the user to the location of the component, the process may be repeated by moving the cursor to the locator view and pressing the left mouse button.

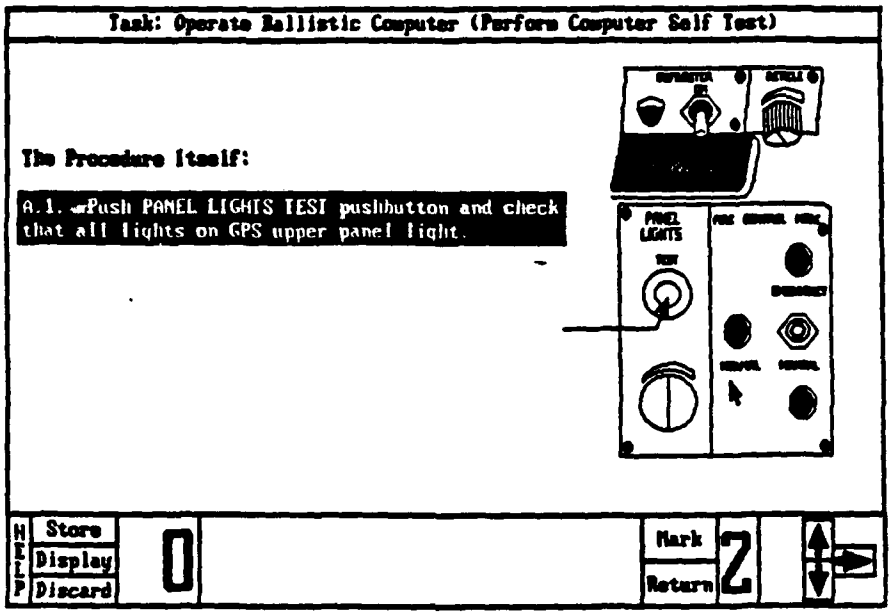


Figure 5-4. Demonstration of Equipment Operation JPA, Screen 4.

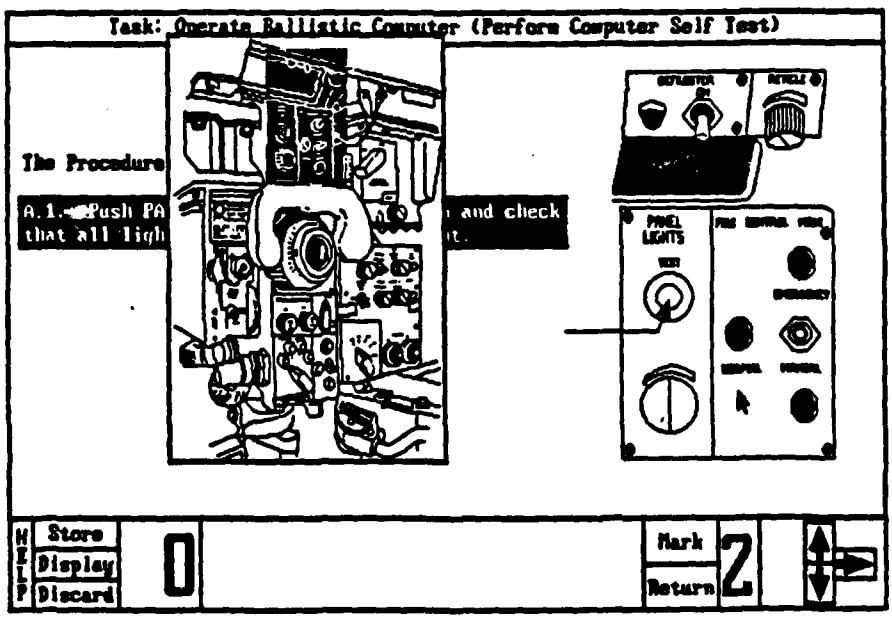


Figure 5-5. Demonstration of Equipment Operation JPA, Screen 5

If the user finds that the detail in a Level 2 screen is adequate, he or she simply places the cursor on the right hand arrow at the bottom of the screen and presses the left mouse button. The next Level 2 screen then appears, as shown in Figure 5-6. In this way, the user can proceed through the database at Level 2.

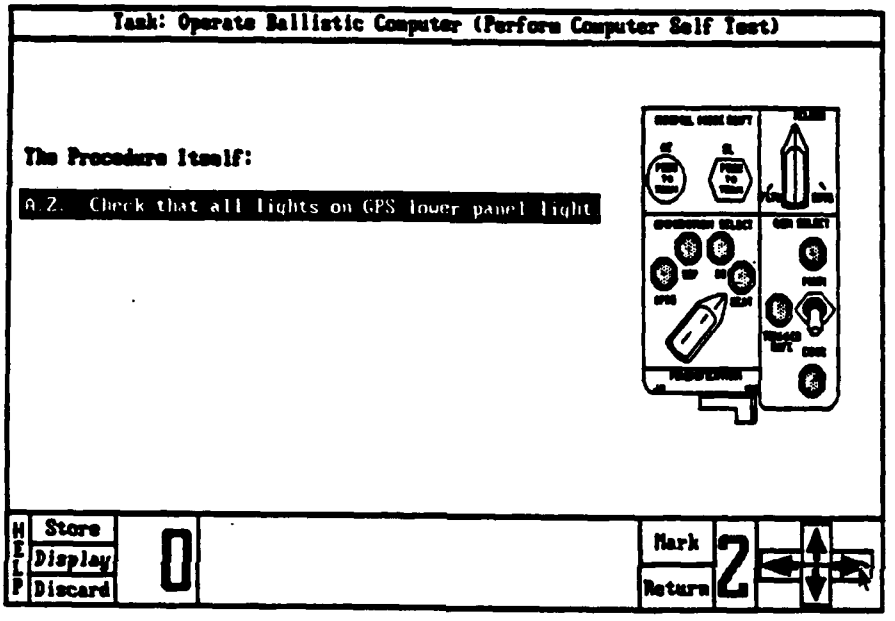


Figure 5-6. Demonstration of Equipment Operation JPA, Screen 6.

If the user intends to browse through the procedure, but wishes to return to this point when finished browsing, he or she may "mark" this screen. To do this, the cursor is placed on the "Mark" zone in the information pane at the bottom of the screen and the left mouse button is pressed. The screen then appears as shown in Figure 5-7. The user may return to the marked screen at a later point in time by placing the cursor on the "Return" zone and pressing the left mouse button.

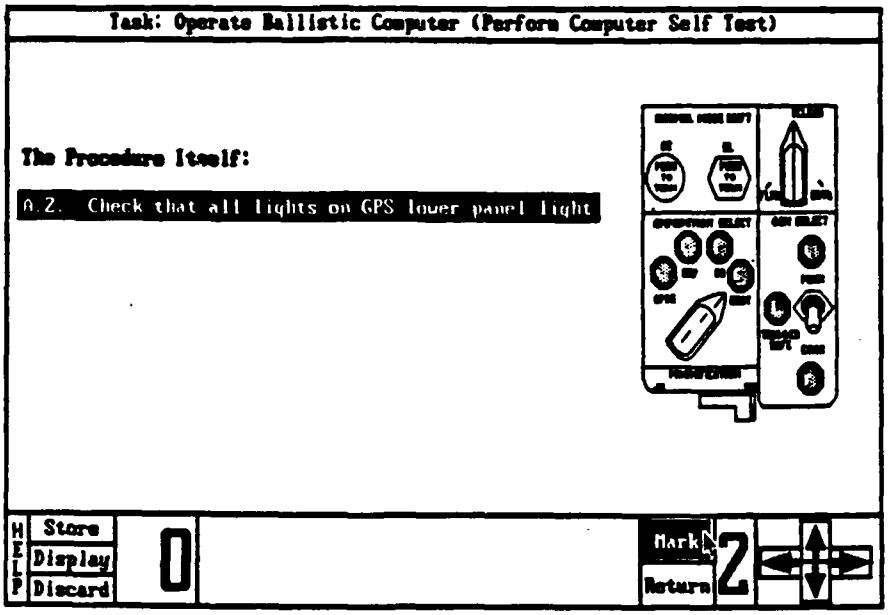


Figure 5-7. Demonstration of Equipment Operation JPA, Screen 7.

At some point in the procedure the user may need more detailed information than that provided by a Level 2 screen. To obtain this information, the user chooses to go to Level 3 by moving the cursor to the down arrow and pressing the left mouse button. A display such as that shown in Figure 5-8 will then appear. When the user moves the cursor down to the message pane at the bottom of the screen and presses the left mouse button, the display changes to a screen which shows how the equipment should look after the specified procedural step has been performed. In the present case, the screen shown in Figure 5-8 would be replaced by that shown in Figure 5-9.

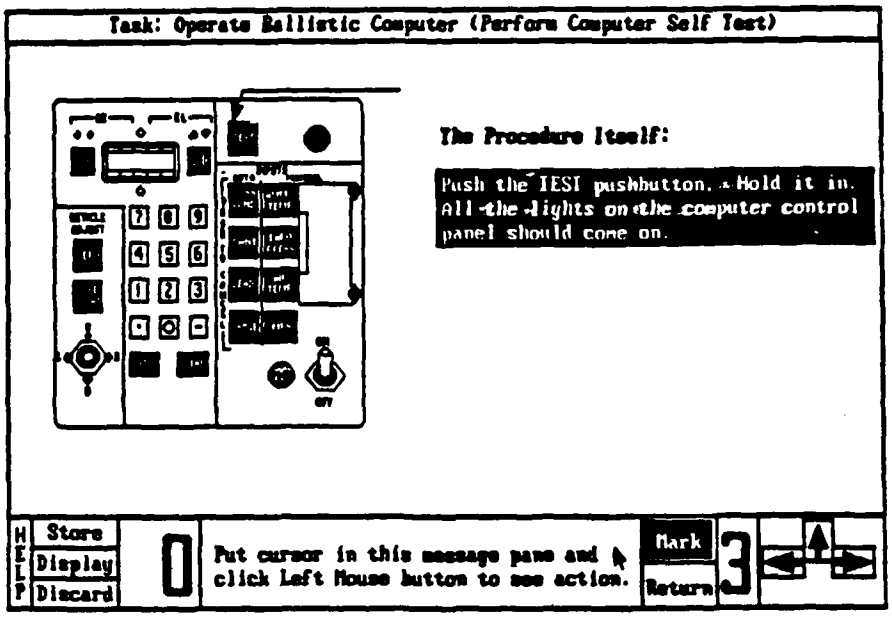


Figure 5-8. Demonstration of Equipment Operation JPA, Screen 8.

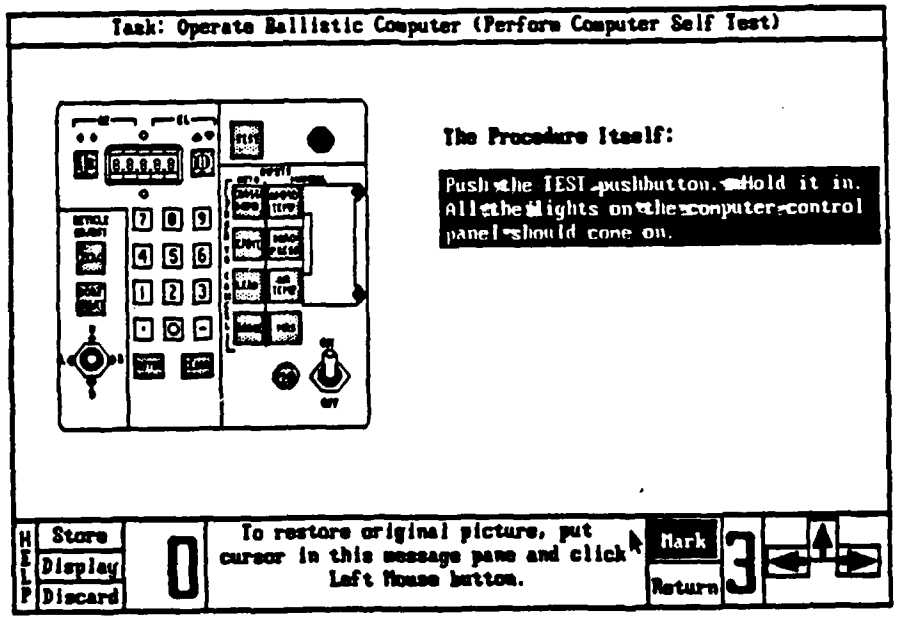


Figure 5-9. Demonstration of Equipment Operation JPA, Screen 9.

Using the four arrows in the information panel, the user can browse through the database. Another way to browse through the database is to call up a map of a procedure (Figure 5-10) and use it to move around the procedure. The map can be displayed by placing the cursor on the screen level number and pressing the left mouse button.

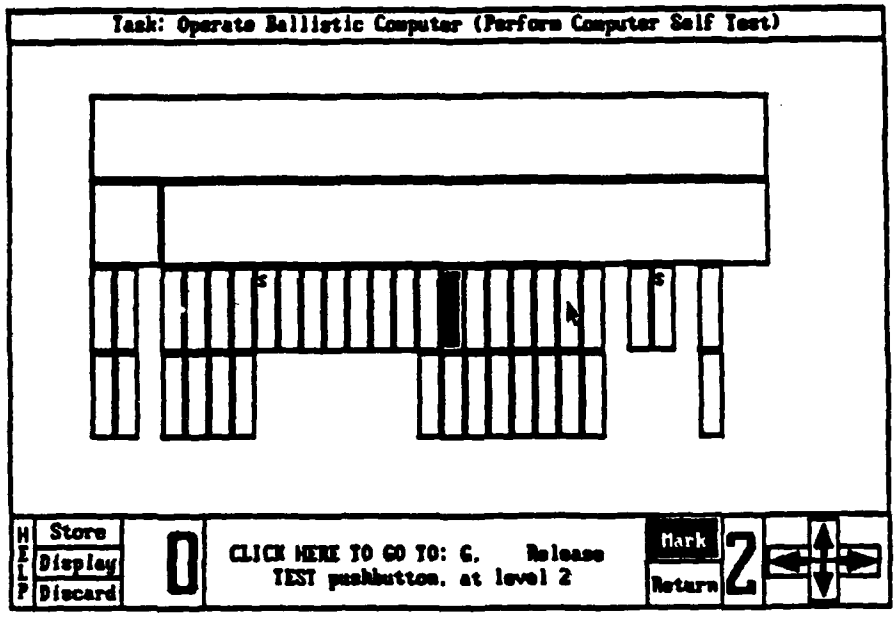


Figure 5-10. Demonstration of Equipment Operation JPA, Screen 10.

A procedure map indicates which screen is currently being displayed, and it indicates the relationship between this screen and other screens. The screen which is currently selected will be represented by a black box in the map, and the words in the information pane at the bottom of the screen describe the procedure text of the procedural step. To jump to any screen shown in the map, the user places the cursor in the box representing the step and presses the left mouse button. Then the user moves the cursor to the information panel and presses the left mouse button. In this example, the screen shown in Figure 5-11 would appear.

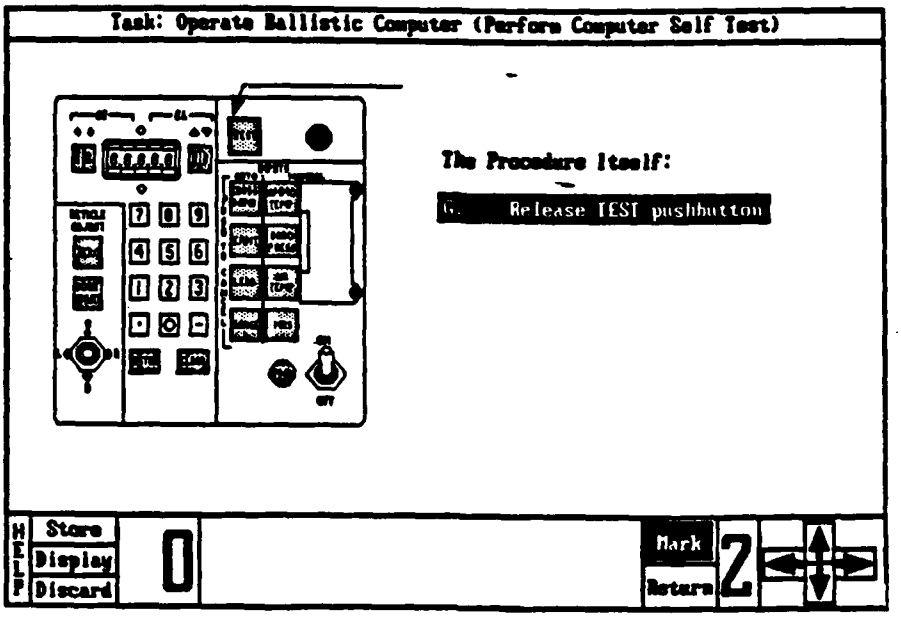


Figure 5-11. Demonstration of Equipment Operation JPA, Screen 11.

The technician continues to familiarize himself with the procedure by browsing through screens which display the steps in the procedure. The student may decide he can perform the procedure with little additional assistance from the screen, except for specific screens containing numbers or other detailed data such as contained in the screen in Figure 5-12. To have quick access to this screen, the technician place the cursor on "Store" and presses the lift mouse button. This places the screen in a special memory. The screen can be recalled by placing the cursor on the "Display" and pressing the left mouse button. A small set of pictures can this stored in this manner. To drop a picture from special memory, the user places the cursor on "Discard" (while the screen to be dropped is displayed), and presses the left mouse button.

After browsing, the technician can return to a marked screen, or review the screens he has stored. He can use the technical information as a job aid (using the screens to guide step by step performance) or as reference information (referring to selected screens, as needed).

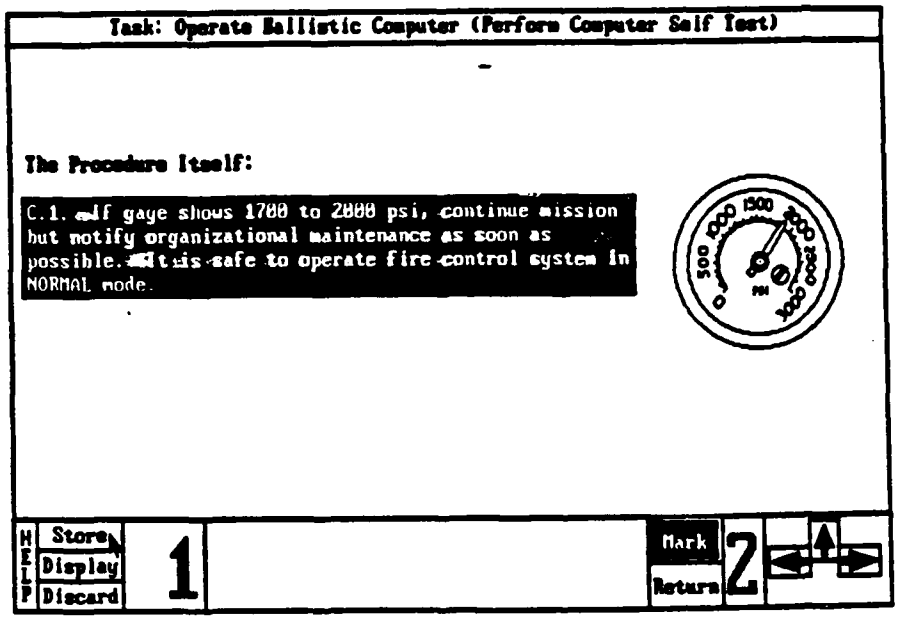


Figure 5-12, Demonstration of Equipment Operation JPA, Screen 12.

Demonstration of Proceduralized Troubleshooting JPA

A brief demonstration of a proceduralized troubleshooting JPA is provided in Figures 6-1 through 6-8. This type of JPA is an extension of the equipment operation JPA demonstrated above. The specific procedure to be demonstrated is the M1 tank gunner task of performing a computer self test. The first screen of this sequence is shown in Figure 6-1. At this point in the troubleshooting procedure, numbers appear in a "seven segment display", and the user is to note the numbers that appear and take action dependent upon the numbers (i.e., go to step N if the number 1 or any combination of the numbers 5, 6 or 7 appear in the display). In this case the number appears in the display. The next two screens, Figures 6-2 and 6-3, provide additional information about the seven segment display and lights on the control panel.

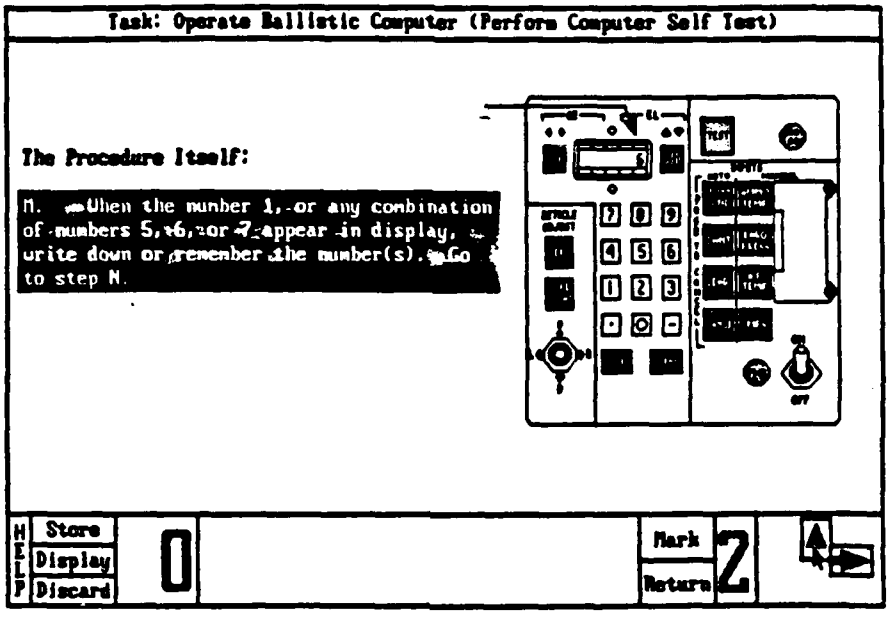


Figure 6-1. Demonstration of Troubleshooting JPA, Screen 1.

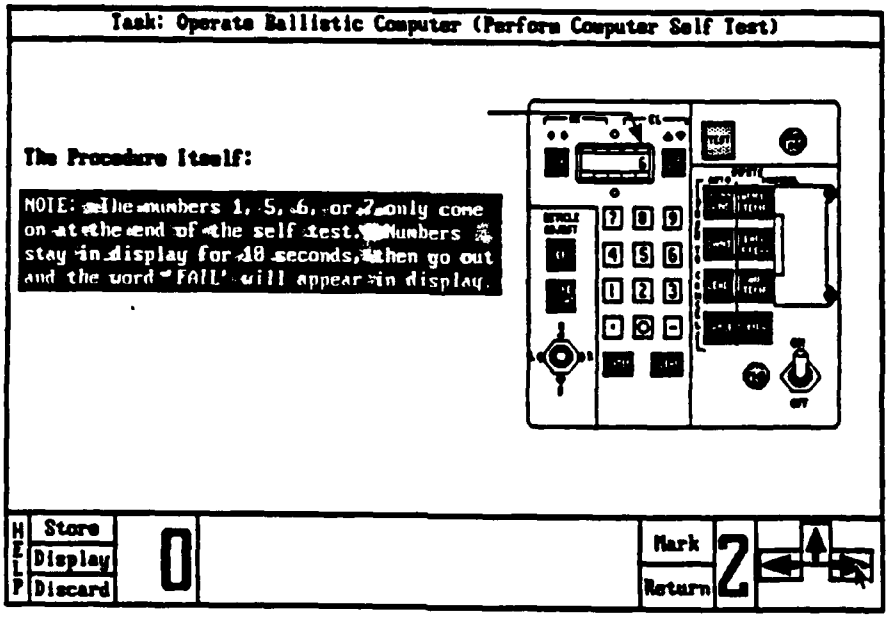


Figure 6-2. Demonstration of Troubleshooting JPA, Screen 2.

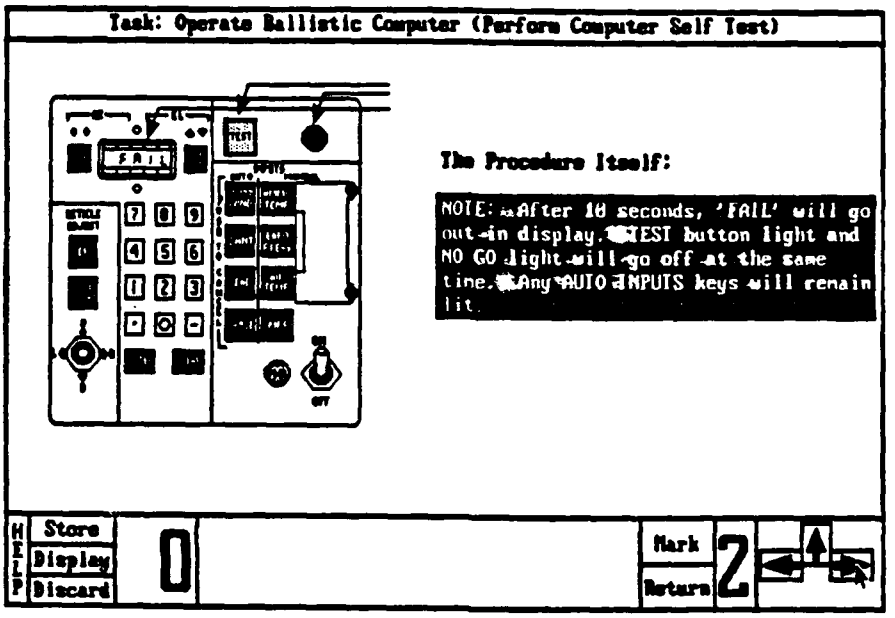


Figure 6-3. Demonstration of Troubleshooting JPA, Screen 3.

The screen displayed as Figure 6-4 notes that if any failure 1 through 8 appears during the self test, then certain corrective actions to be specified should be taken. For this demonstration, assume that a three (3) appears in the seven segment display. In scanning through the next few frames, the information in Figure 6-5 is encountered. It states that if failure 3 appears, the operator is to troubleshoot the crosswind sensor. By placing the curser on the down arrow and pressing the left mouse button, that topic is selected. Then by placing the cursor on the down arrow and pressing the left mouse button, the program jumps to the procedure of troubleshooting the crosswind sensor. Figure 6-6 shows the first few steps in troubleshooting the crosswind sensor.

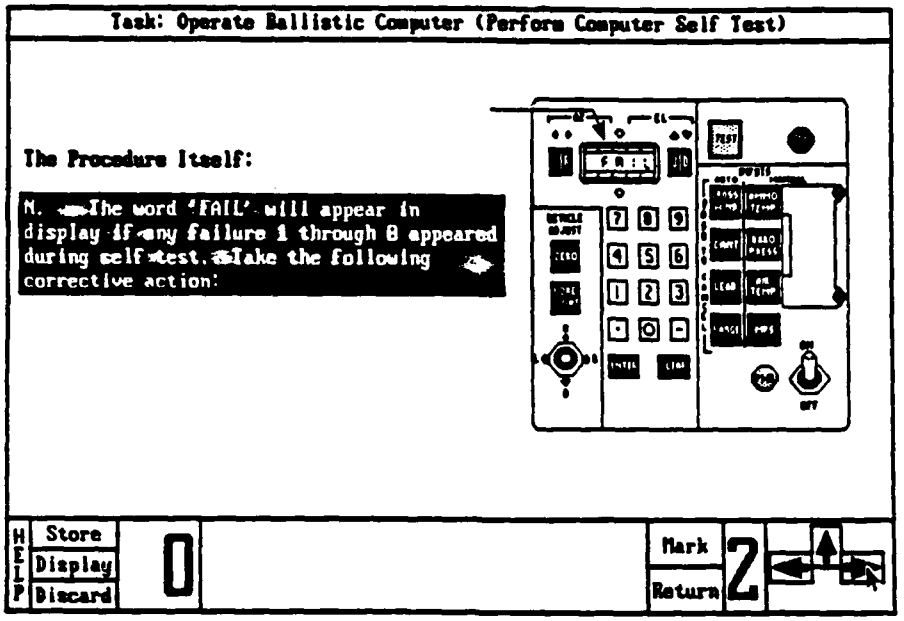


Figure 6-4. Demonstration of Troubleshooting JPA, Screen 4.

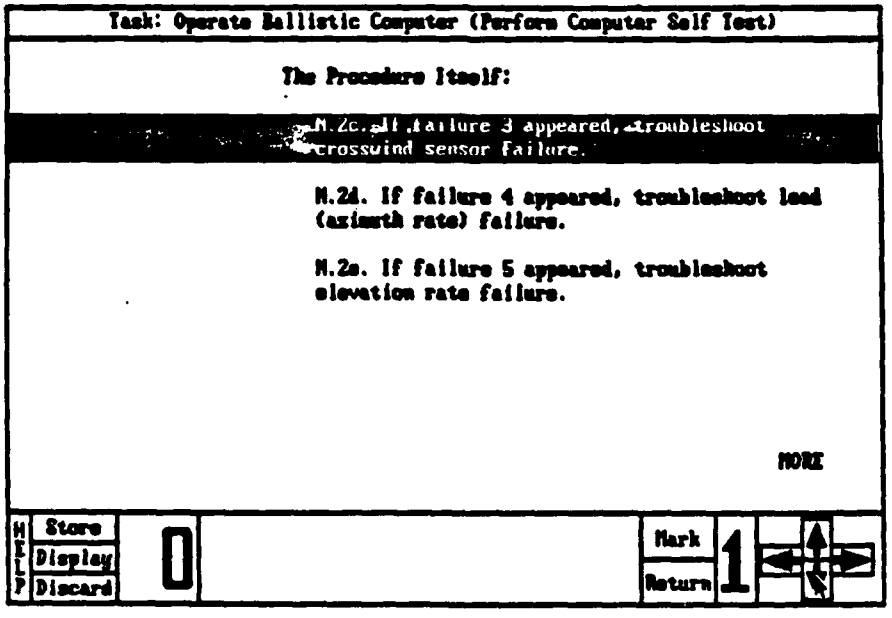


Figure 6-5, Demonstration of Troubleshooting JPA, Screen 5.

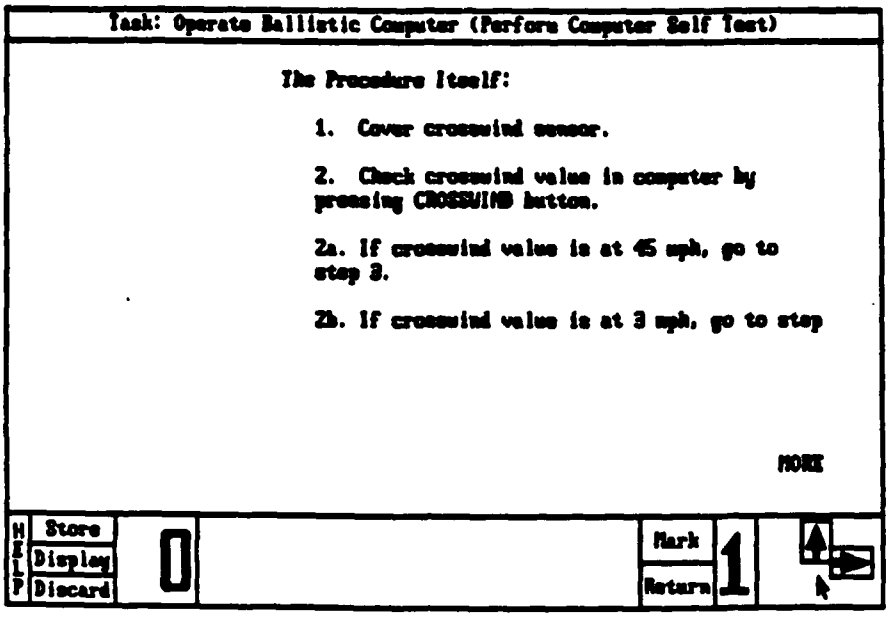


Figure 6-6. Demonstration of Troubleshooting JPA, Screen 6.

To obtain more information on any individual step, the cursor is placed on the text of that step, and the left mouse button is clicked. Then the down arrow is selected to display the detailed information. In this instance, step 1 is selected and Figure 6-7 displays the next level of information on that step.

Additional information can be displayed to show where the component is installed on the M1 tank. When the student places the cursor on the graphic and presses the left mouse button, an overview of the tank is displayed indicating where the crosswind sensor is located. Figure 6-8 shows this display.

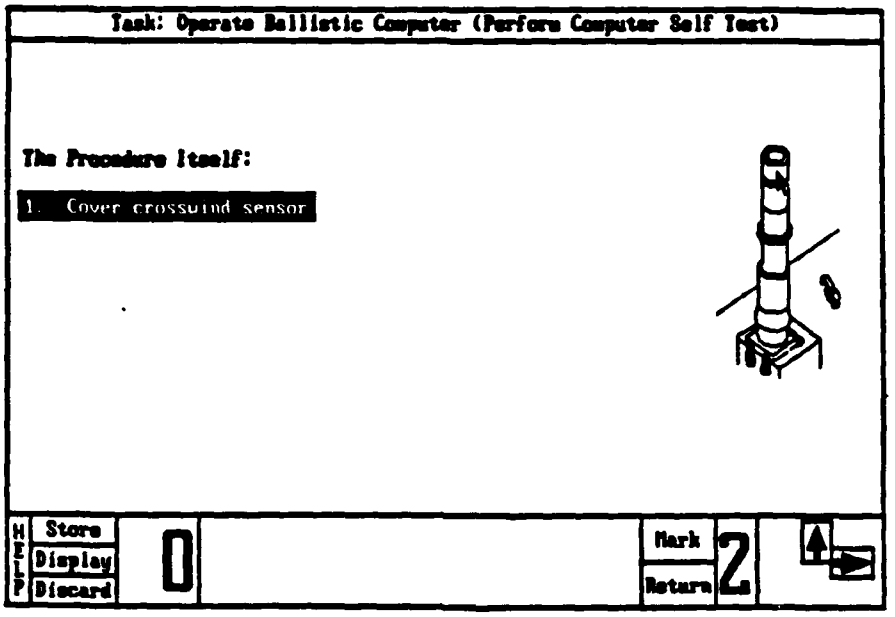


Figure 6-7. Demonstration of Troubleshooting JPA, Screen 7.

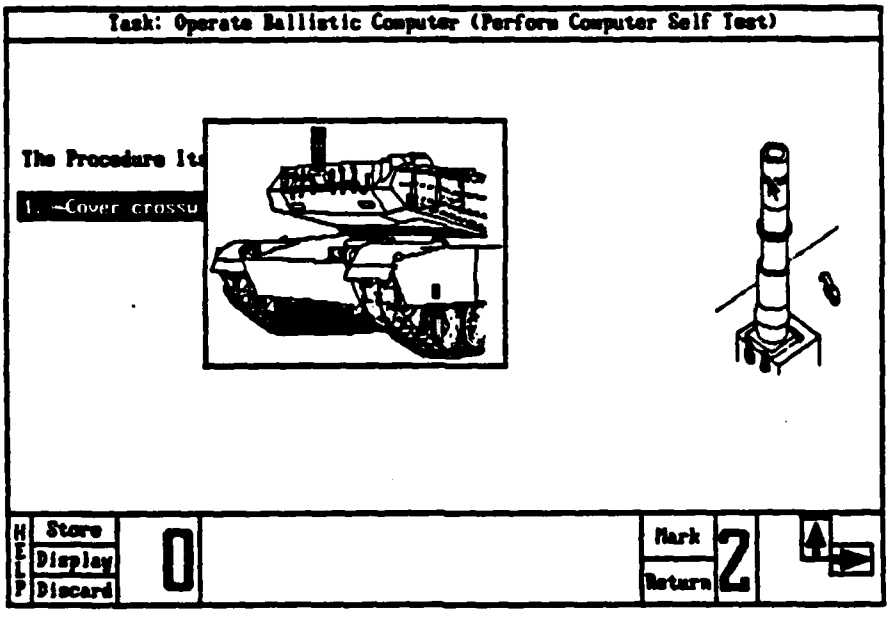


Figure 6-8. Demonstration of Troubleshooting JPA, Screen 8.

Authoring Job Performance Aids

The emphasis of this project was screen format designs and display algorithms rather than authoring system development. However, authoring tools were needed in order to create the demonstration screens. These tools were developed in Smalltalk, as described in the section titled "Programming Approach." The screens and authoring system were demonstrated using a demonstration software package titled "Procedure Job Aid Production System (PJAPS)".

The resulting system serves the immediate purpose and provides the basic architecure for a fully-developed authoring system. There was no attempt to make the tools user friendly, and initially, there was no attempt to document the use of the tools, since this goals would be outside the scope of the task and available resources.

A user's guide for the demonstration software, addressing the information delivery and authoring systems, was subsequently developed by the government with the assistance of Eagle Technology. The User's Guide for the PJAPS demonstration software is provided in Appendix D. The primary authoring system functions for the PJAPS are described on the following pages.

Create/Modify Procedures. This function is used to enter or update the steps, graphics, and common subprocedure links for each procedure. The author types in each step, then indicates which graphics is associated with each step. When specifying graphics, the author is also asked to designate the points of interest (i.e., areas on the figure to be emphasized with arrows). If the graphic is animated, the author must also specify which of the two figures is the initial view. Other modifications allow authors to make modifications to existing procedures. Figure 7 shows the screen during the development of the Computer Self Test procedure.

Create/Modify Static Graphics. This function allows the author to draw or change the line drawings displayed in the PJAPS. A variety of graphic tools are provided, such as the ability to generate squares, circles and other geometric forms. In addition, a zoom editor is available which allows the author to enlarge an area of the drawing for more precise work. Figure 8 shows the zoom editor in use.

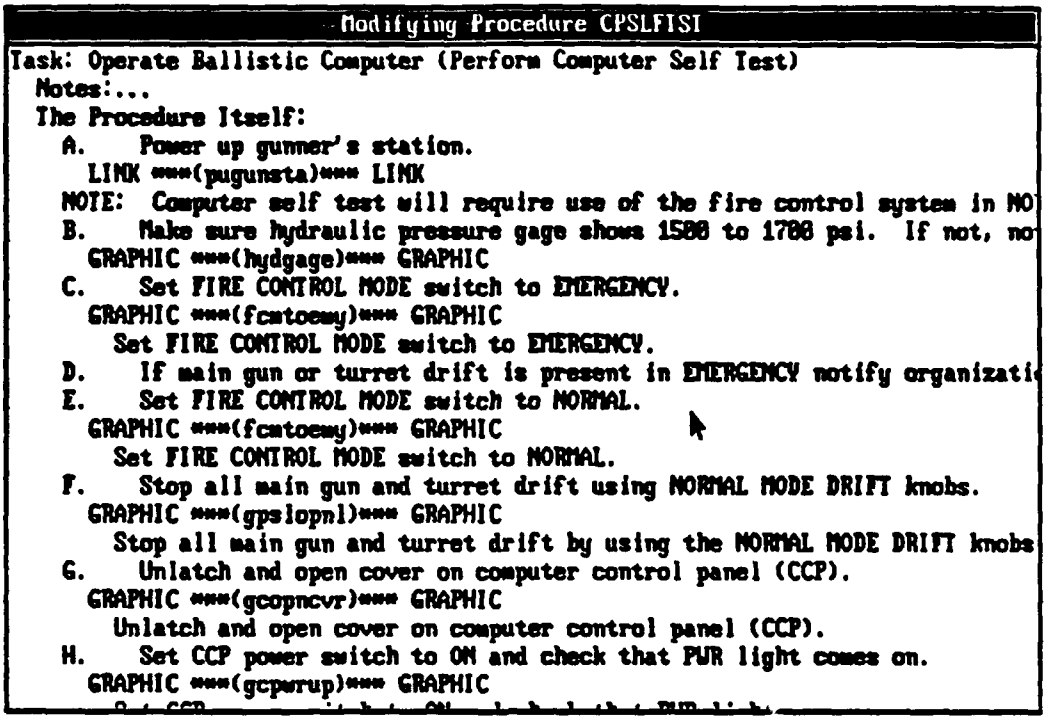


Figure 7. Create/Modify Procedure Screen.

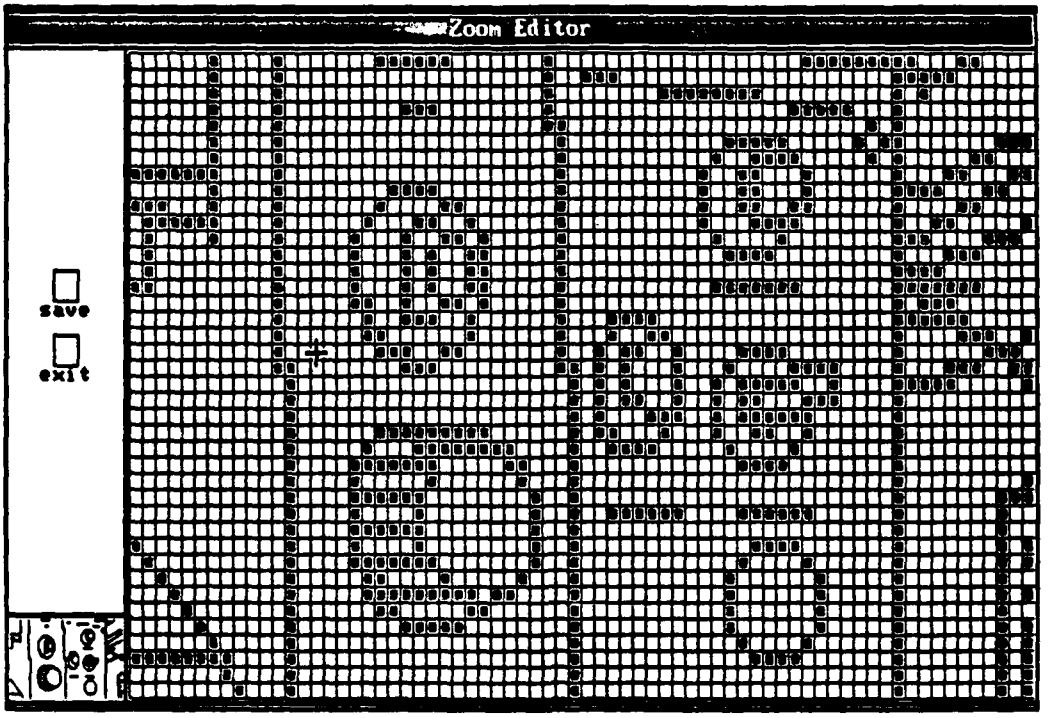


Figure 8. Create Modify Static Graphics-Zoom Editor.

Create/modify Animated Graphics. This function uses the same editing features as the static graphic function, but in this case the author is working with two views of the same picture. The author will normally construct one view and then, using the copy function, create an identical second view. The second view is then edited to reflect the results of the action. For example, a switch may be changed from the off position to the on position. Figure 9 shows the two panes used in this editor.

Import Scanned Art. While this is not a function in the same sense as the others, the use of scanned art is an important capability of the system. Using a scanner and commercially available software, drawings from technical manuals and other sources can be scanned and transferred into the authoring environment. Once there, such pictures normally require clean-up due to the differences in resolution between the screen and printed page. Much of the clean-up work is accomplished using the "zoom edit" function of the system. The scanning process, while not instantaneous, is still more efficient than drawing the graphics completely freehand. The left portion of Figure 10 shows a drawing as it might look after being scanned in the system, and the right portion shows the same drawing after clean-up.

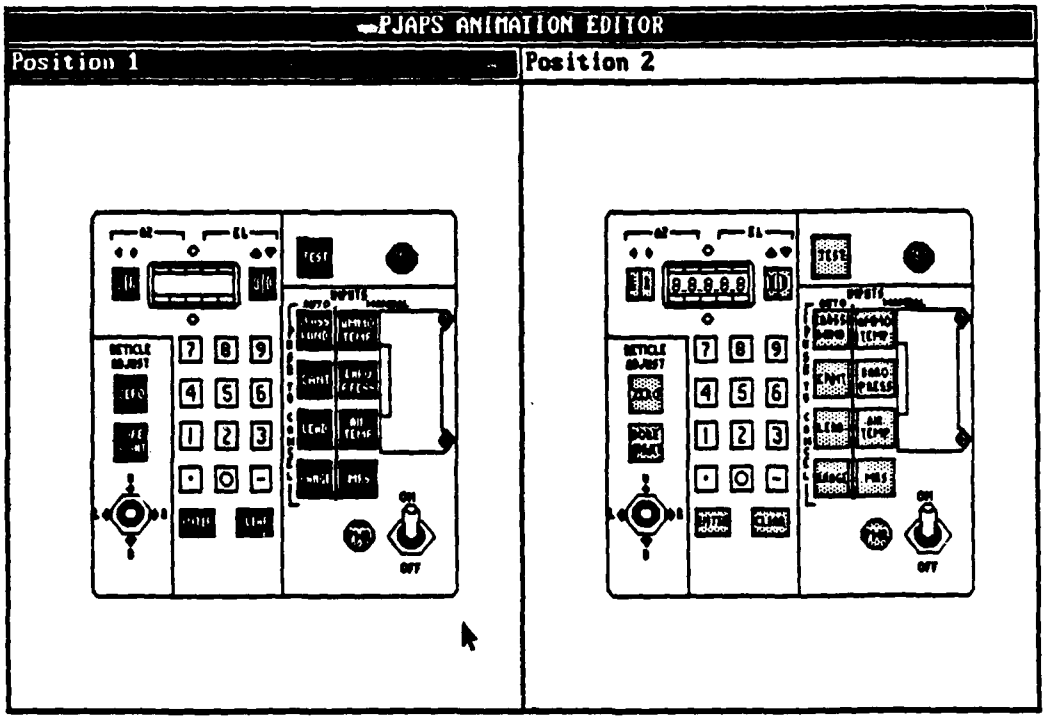


Figure 9. Create/Modify Animated Graphics

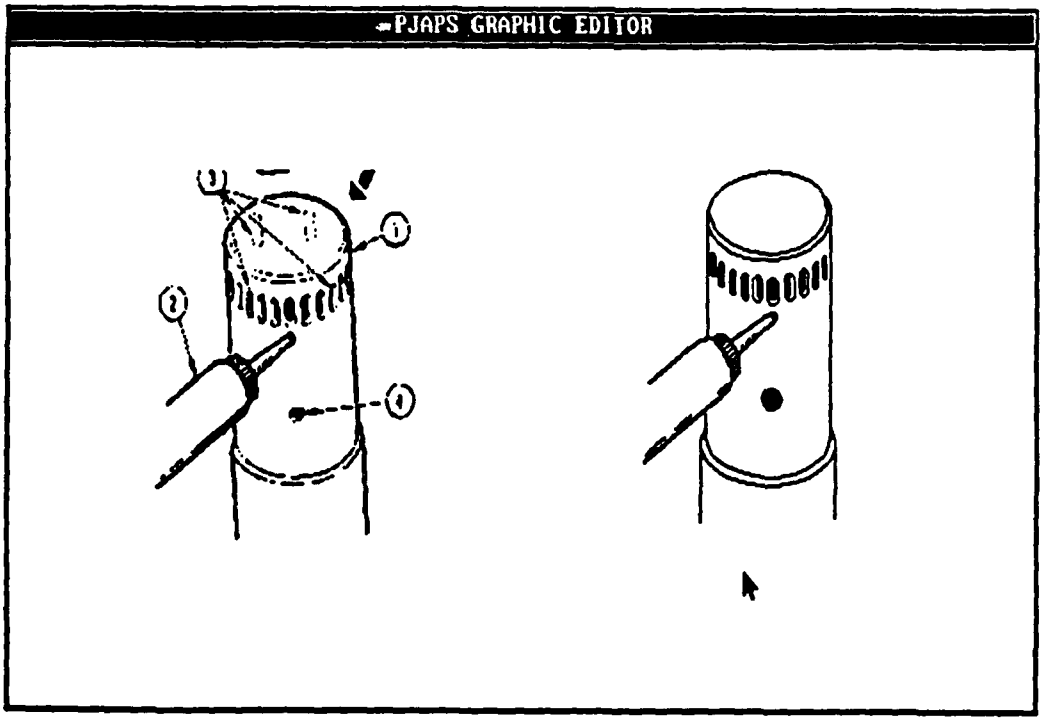


Figure 10. Example of Imported Scanned Art

Page 42 Blank

SECTION IV:
FORMATS FOR PROCEDURE SIMULATION

Broad Goals of Procedure Simulation Formats

Formats for procedure simulations must include a simulation component and an instructional component. Procedure simulation screens present a dynamic graphic model of an item of equipment that can be manipulated by a student to perform a specific procedure. For example, switches can be repositioned and valves opened or closed. In response to these actions, the graphic replica of the equipment performs realistically (e.g., lights go on or off, gages move). One might imagine, for example, how such a simulation might provide a student with practice identifying and troubleshooting problems with equipment.

In addition, a variety of instructional features must be present to guide the student's practice by, in part, providing messages at appropriate moments on the computer display. These features include providing explanations of the tasks to be performed and feedback regarding the adequacy of performance.

Meeting the simulation and instructional goals described above in a cost-effective manner requires "smart" computer screens. The computer screens are called "smart" in that the logic for performing dynamic screen operations is built into the display algorithms. The formats for procedure simulations must account for all the "smart" capabilities of the display screen.

In the discussion of the formats for procedure simulations, definitions of terms will prove useful. In this report, "smart object" refers to a manipulatable object on the computer screen, including all the graphic representations of the object and the computer code that make it possible for the on-screen object to behave like the object it represents. Included in this code are all the messages the object sends, under specified conditions, to other objects and the capability to receive and respond to messages from other objects.

A "smart format" refers to the design of a class of screen to support a specific type of function. This design incorporates smart objects. Included within the format design are the basic layout of the display and the interactions with students and other screens.

A "smart screen" is the result of implementing a smart format. The smart screen includes text and graphics of a procedure, objects to be manipulated, and feedback on the results of the manipulation. A smart screen may incorporate randomness, step ordering, and displays, based on the current mode of the system. All this takes place in accordance with the specifications contained in the smart format.

Procedure Simulation Functions

This section describes the general characteristics of the eight types of smart formats created by procedure simulations. Depicted here are the dynamic functions performed by screens, either as a response to student input or by direct control of the computer. After descriptions of the screens are presented, a sequence of formats will be presented to demonstrate how the formats used to create a flow of events similar to practicing on equipment. This is followed by a brief description of the authoring routines used in creating a procedure simulation incorporating these smart formats.

Each procedure simulation display format incorporates some or all of the functions listed below.

- Use of Text and Static Graphics. The equipment to be operated can be represented as line art, or as continuous gray scale pixel art. This can be presented in black and white or color.

- Knowledge of Steps. Computer code stored with the equipment graphic contains data on the procedural steps that can be performed on the graphic.

- Knowledge of Sequence. The computer code contains data on the acceptable sequence of steps, including alternatives and a random order of steps if no fixed order is required.

- Knowledge of Prior Actions. When procedure options and multiple paths through a procedure are acceptable, the screen retains a knowledge of past actions so as to properly present equipment conditions.

- Knowledge of the Behavior of Smart Objects. When a smart switch, for example. is used on a screen, the switch must carry code with it to display all positions of the switch. Information on the visual representation of all smart objects is stored in this manner.

- Knowledge of Dependencies. The code associated with smart objects must contain messages to be sent to other objects on the screen that are affected by the condition of the smart object. For example, when a switch is placed in a new position, messages must be sent to all components that are dependent upon the switch position.

- Control of Timed Events. The code for a display screen contains data on delays (how long until a message should appear) and rates (how many seconds between light flashes).

- Display of Screens in Various Modes of Operation. Data is contained in the screen display so that the screen can be configured in terms of content and layout for modes such as initial practice and advanced practice.

- Selection of Random Events. The computer code associated with a screen must contain a capability, when required, to randomly select its condition from among a set of alternatives. Consider the case of a troubleshooting procedure where a student might make a differing diagnosis of a problem, depending on the specific findings of equipment checks. This function makes it possible to randomly vary the specific equipment components that are malfunctioning and provide a greater breadth of practice in troubleshooting procedures.

- Monitoring of Student Inputs. The code must detect student inputs, including correct, expected incorrect and unexpected behavior.

- Response to Student Inputs. The code associated with a screen must respond to student inputs by generating changes to the graphic representation of the equipment, text messages or in some other way to provide appropriate feedback to the student on actions taken.

These functions, as required, are built into the eight "smart" formats used in creating procedure simulations. Each format is described below, along with the functions built into it.

Formats for Procedure Simulations

Text/Graphic Screens. There is a general class of screens that contains static text and/or graphics. When this class of screen is used the author has almost complete freedom in the use of the screen, in terms of text or graphic displays. Two functions are automatically included in each of these screens. First, line 1 of the screen contains the name of the procedure. Second, line 24, the last line, contains the message "Press ANY MOUSE BUTTON to continue". The code to respond to this message is also automatically included. See Figure 11 for an example of a text screen.

Menu Screens. Menu screens require the student to choose a course of action. At least two items must be present in each menu. Selecting an item from the menu causes the program to jump to the designated program element. Certain functions are automatically included in a menu screen. Line 1 of the screen contains the name of the procedure. The menu items are active and change to reverse video when the curser rests on them. The last line in the menu, by convention, is used for the "exit procedure" option, which automatically returns the program to the previous menu. Instructions for selecting a menu item are included at the bottom of the screen. Selecting a menu item sends the program to the next appropriate display. See Figure 12 for an example of a menu screen.

Figure 11. Text/Graphics Screen.

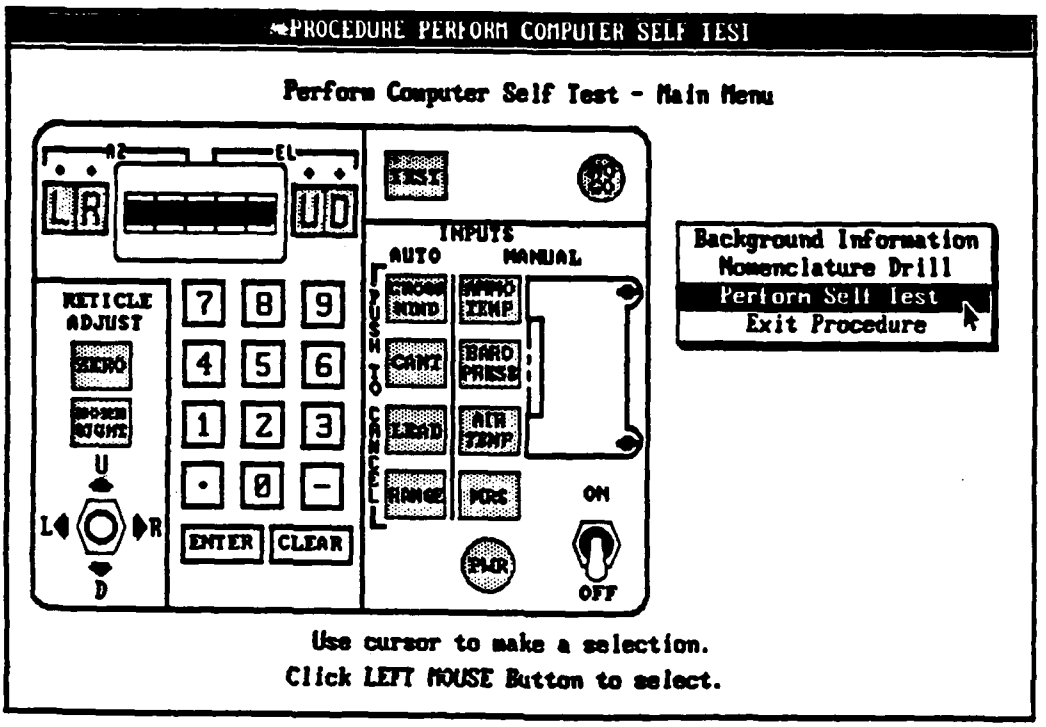


Figure 12. Menu Screen.

Invisible Menu Screens. This format is used for a menu when automatic jumps to randomly selected destinations are required in advanced practice. It allows the author to determine the probability for each alternative to be selected. In initial practice, the menu is visible to the student, and a menu item must be selected manually by the student. See Figure 13 for an example as it appears to the student in the initial mode and Figure 14 as it appears to the author, indicating the probability of any specific failure being introduced. The last line in the menu is to Exit Procedure, which sends the program to the previous menu.

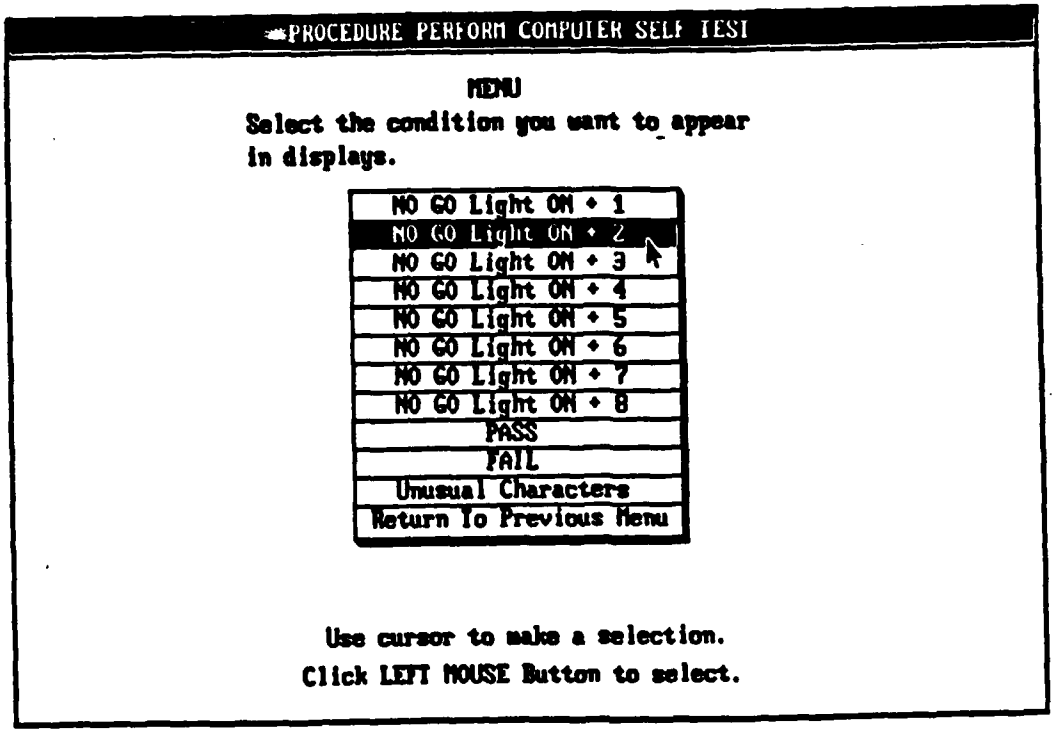


Figure 13. Invisible Menu Screen—Initial Practice

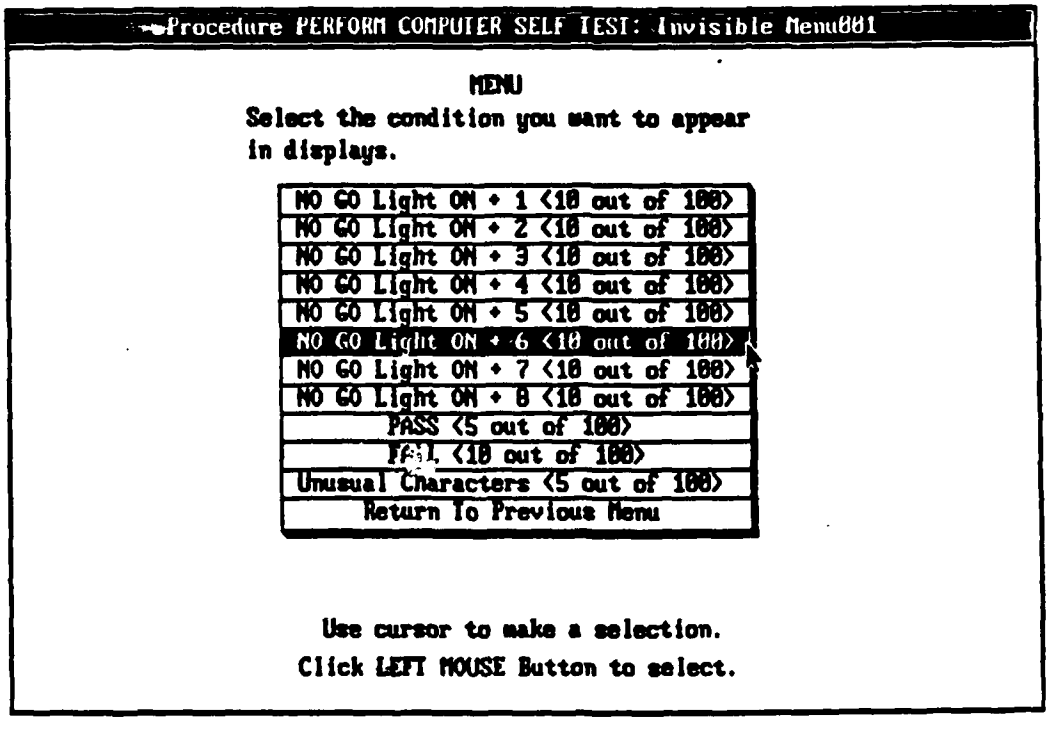


Figure 14. Invisible Menu Screen—Advanced Practice

Locator Screen. The locator screen (see Figure 15) displays an overview of the equipment and serves as a graphic index to the close-up views needed to perform a step in the procedure. One or more zones on the overview are "hot". These zones appear the same as any other area of the screen, but when the cursor is placed in one of these zones, and the mouse button is pressed, an action screen with a close-up view is displayed.

As in previous screen formats, line 1 is reserved for the name of the procedure. Three panes make up the rest of the screen. The top pane displays the text of the procedure step. The large middle pane is for the graphic overview and any text messages needed to direct student input. The bottom pane is a message area used to display a time-delayed message giving directions to the student on how to operate the locator screen. It is also used to present feedback on correct or incorrect responses. In terms of Figure 15, if the student selects the wrong location for the hydraulic gauge, the message "No. Try again" appears in the bottom pane.

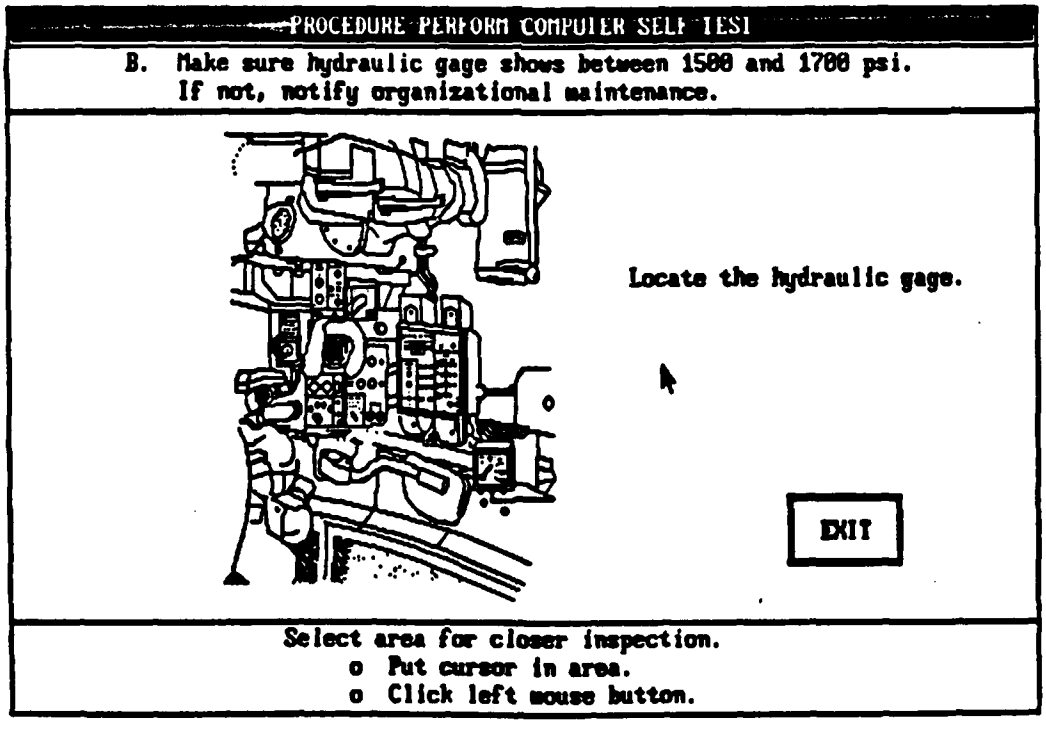


Figure 15. Locator Screen

Action Screen. The action screen is where a student performs a procedure. It presents the dynamic graphic of the equipment on which a specific step in the procedure is performed. It is not a free play model, but rather a limited model on which only the appropriate step can be performed. All other actions are evaluated as incorrect actions. After an error, the student is coached on the correct action. In initial practice, the correct object to be located or manipulated is flashed after the first error (i.e., quickly shown in reverse video), and in advanced practice the correct object is flashed after the third consecutive error. The only way to advance from this screen is to perform the procedure correctly.

Line 1 of the screen contains the procedure name, and the pane below displays the text of a procedural step. The large middle pane is for the graphic model of the equipment and is where the action takes place. The bottom pane provides feedback messages to the student. See Figure 16 for an example of an action screen.

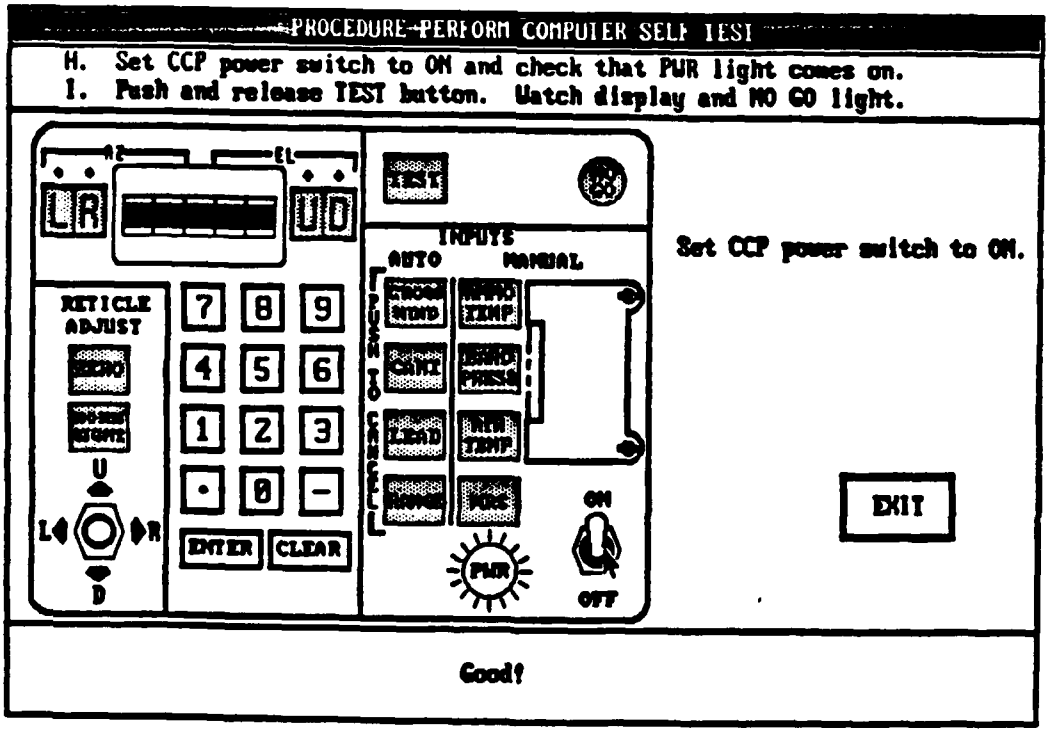


Figure 16. Action Screen

Training Strategy Screen. The Training Strategy Screen is a single function preprogrammed screen. It allows the student to repeat practicing a segment of the procedure, and it is displayed only in the initial practice mode. See Figure 17.

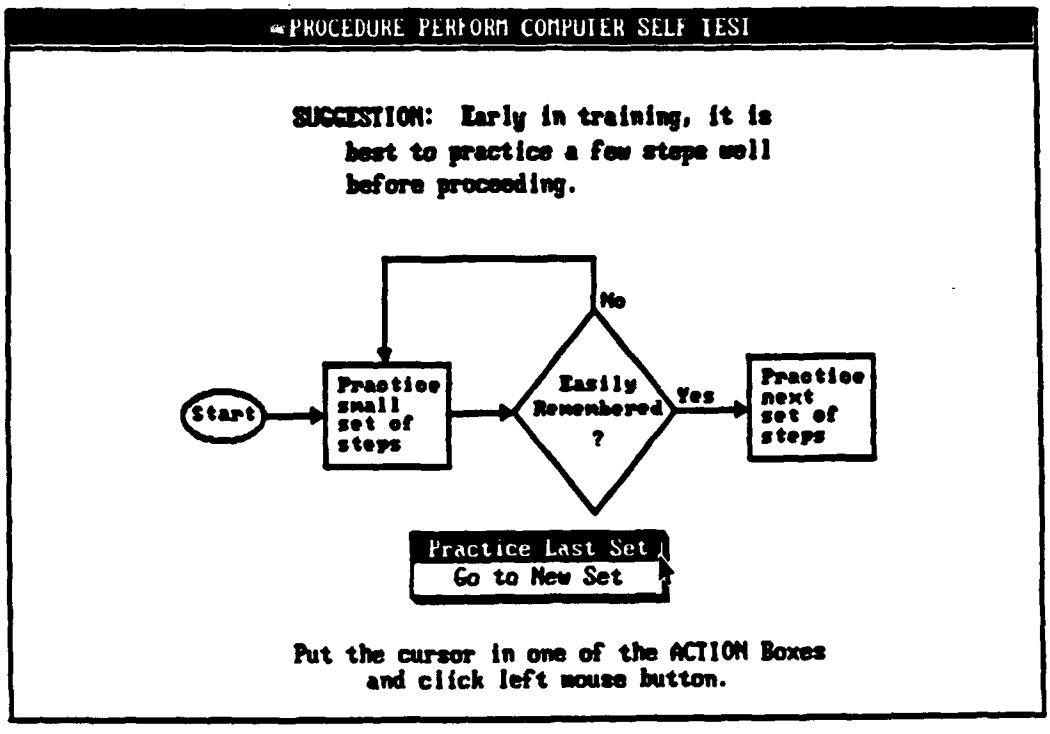


Figure 17. Training Strategy Screen

Note Screen. The Note Screen provides a break in the procedure simulation, allowing the student to practice a specific operation before performing that operation within the simulation. It is like the Action Screen except that the student can continue to practice an operation as long as he chooses, and the actions are not a part of the simulation and are not scored. The student must activate a "continue button" to jump back into the simulation program. This type of screen is used only in initial practice.

Line 1 of the screen contains the procedure name, and the pane below contains the text of the procedural step. The large middle pane contains the directions for practice and the background objects and smart objects on which the practice is performed. The bottom pane displays feedback messages and time-delayed directions on how to operate this pane. It also contains the "continue button" to be used in jumping back to the simulation. See Figure 18.

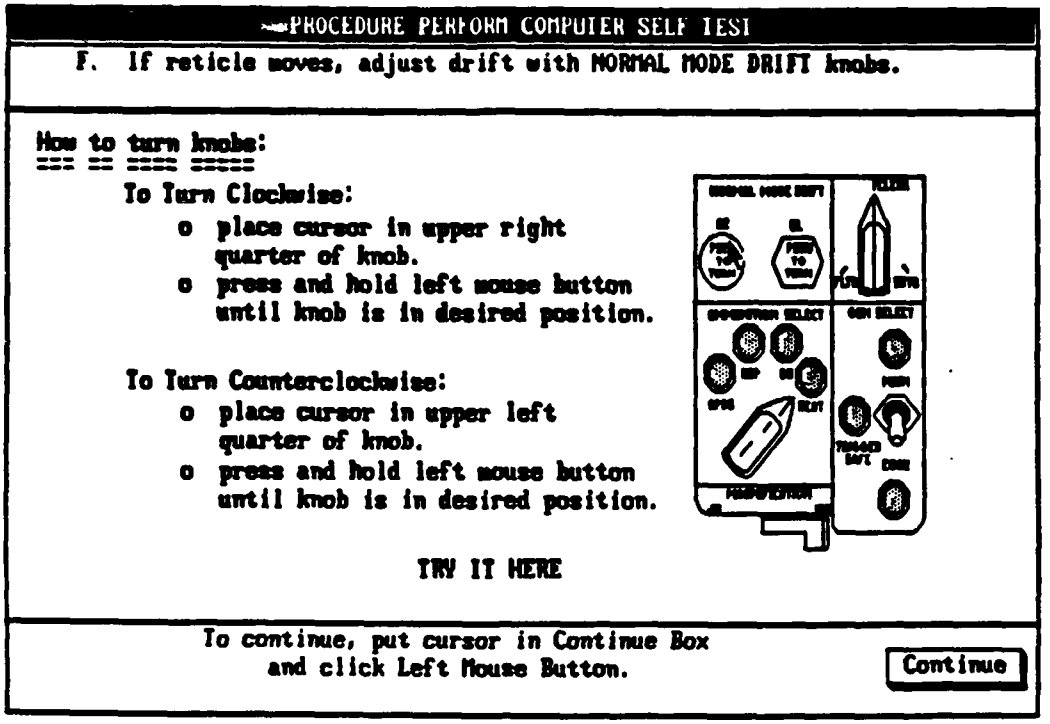


Figure 18. Note Screen

Mode Screen. The Mode Screen shown in Figure 19 is a single function preprogrammed screen which allows the student to choose Early Practice or Advanced Practice. When a practice mode has been selected, the code within this format causes various parameters to be set in other parts of the program.

When early practice is selected:

- Text in the step pane is ON (displayed)
- Note screens are ON (used)
- Training strategy screens are ON (available)
- Invisible menu screens are VISIBLE (student sees menu and must select option).

When advanced practice is selected:

- Text in the step pane is OFF (step does not appear unless pane is pressed)
- Note screens are OFF (not displayed).
- Training Strategy Screens are OFF (no capability for repeated practice of individual steps)
- Invisible menu screens are INVISIBLE (jumps in program made automatically based on probabilities established by the author).

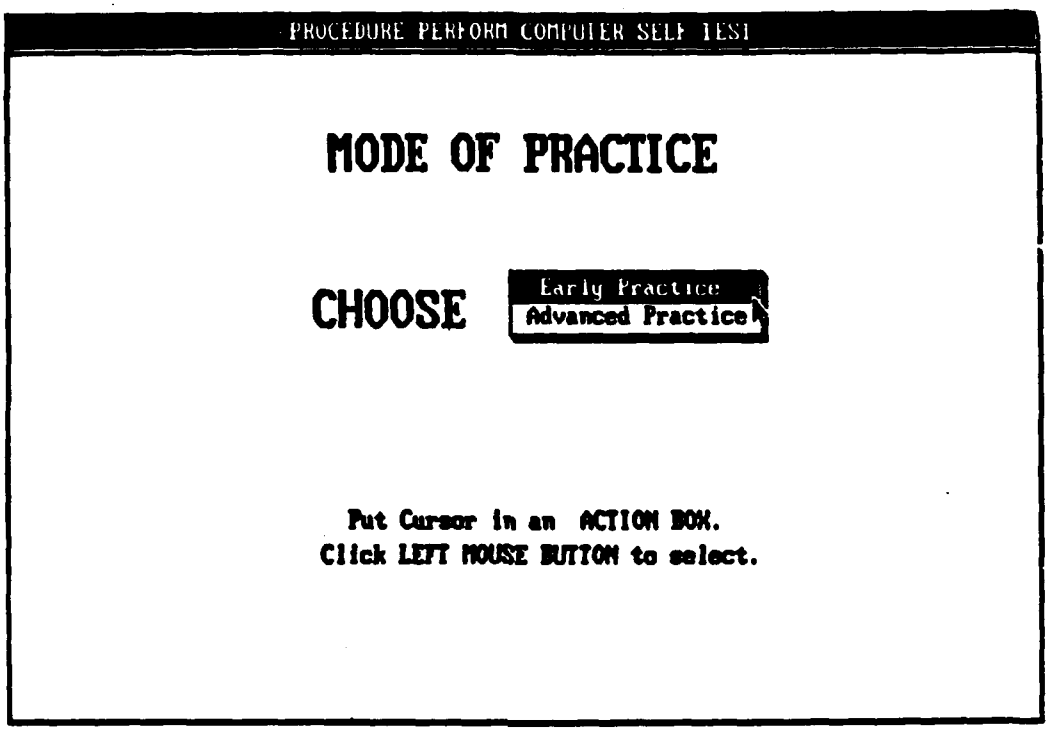


Figure 19. Mode Screen

Sequencing the Formats. Individual formats are the building blocks of a procedure simulation. Constructing a procedure simulation with these building blocks entails establishing a sequence of these formats. The sequencing of events in a procedure simulation was carefully studied by Thomas, Braby and Mears (1988). The flowcharts from that study appear in Appendix A and document a sequence of events for initial and advanced practice of procedures that proved effective in field trials of procedure simulations. It is this sequence that is followed in the current effort.

Demonstration of a Procedure Simulation

A brief demonstration of a procedure simulation is presented in Figures 20-1 through 20-18. This demonstration consists of a sequence of paper graphics of computer screens and depicts screens in the procedure simulation developed in this project. For a proper demonstration of this electronically displayed situation, the dynamic computer displays developed in the Procedural Simulations Produced by Rehearsal (PROSPR) demonstration software must be experienced.

The procedure being simulated is the computer self test on the ballistic computer at the gunner's station of the M1 Tank. This is the same procedure supported with an electronic JPA.

The procedure simulation provides students with a medium for practicing a procedure. It should be noted that the program is intentionally limited to this single type of objective. It does not describe a procedure, as in a JPA. Nor does it teach the student to perform the steps of the procedure, as in a tutorial computer assisted instruction lesson. However, in supporting student practice and by incorporating random failures and in adapting to the skill level of a student, the program has an extended useful life in terms of the number of times a student will use the program to increase skill (i.e., use it to increase speed, accuracy and ability to adapt to real time contingencies).

The "System Configuration" screen is one of the first screens presented to the student. It establishes the initial conditions of the equipment being simulated. Figure 20-1 shows the contents of such a screen. This is followed by the "Mode of Practice" screen, as seen in Figure 20-2. In this instance the mode selected is "initial practice". In selecting "initial practice", the program automatically provides certain guides and prompts needed by beginners and allows the student to repeat the practice of the more difficult parts of the procedure. It is not scored.

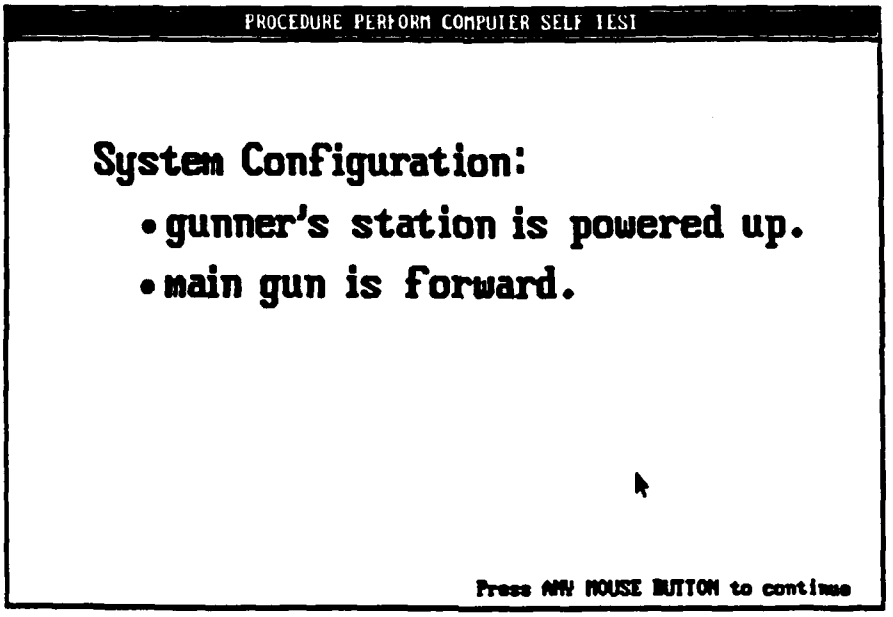


Figure 20-1. Demonstration of a Procedure Simulation, Screen 1.

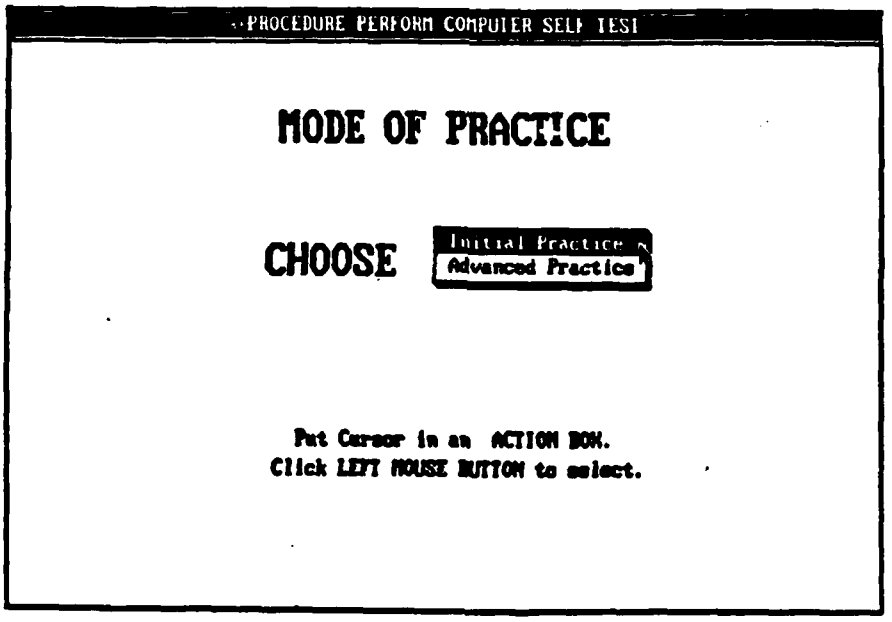


Figure 20-2. Demonstration of a Procedure Simulation, Screen 2.

The procedural step to be trained and practiced in the next few examples is that of checking a hydraulic gauge to ensure that it shows the proper reading (1500 to 1700 PSI). The screen presented as Figure 20-3 asks the student to locate the hydraulic gauge by placing the cursor in the correct area and pressing the left mouse button. If the student is unfamiliar with the gunner's station and selects the wrong area, the message "Try Again" will be displayed at the bottom of the screen and the correct area will be flashed (i.e., quickly displayed in reverse video). After the student selects the correct area, a close up view of the hydraulic gauge is provided as shown in Figure 20-4.

The new screen allows the student to read the gauge and select an appropriate course of action. In this instance, the pressure is too high, and the student has selected to notify organization maintenance. After three seconds, "Good" is displayed at the bottom of the screen as feedback to the student.

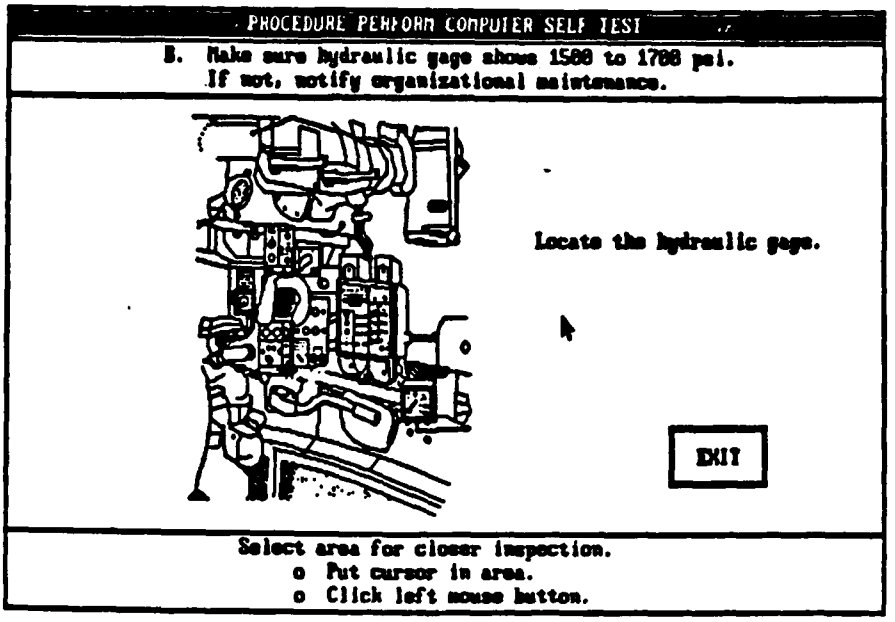


Figure 20-3. Demonstration of a Procedure Simulation, Screen 3.

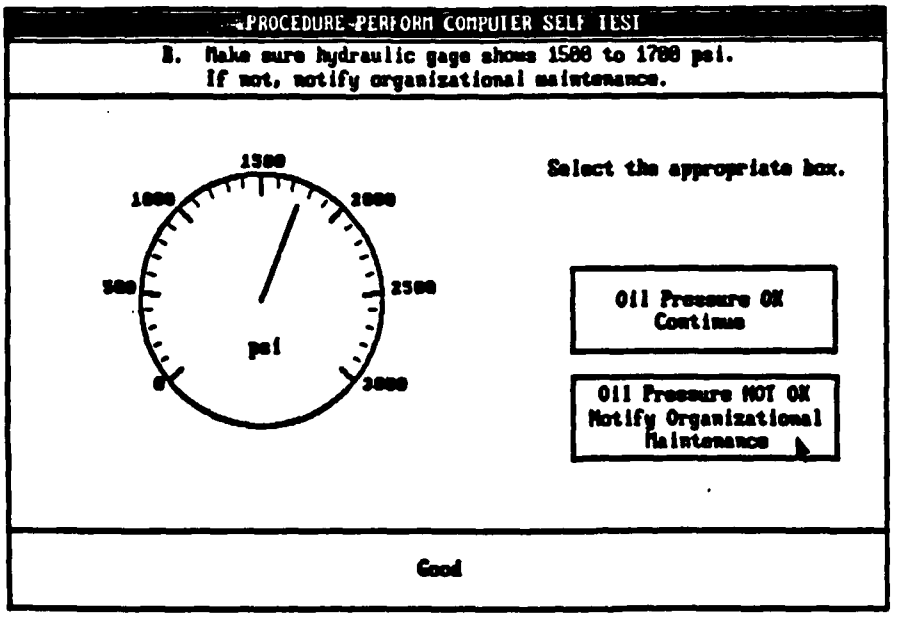


Figure 20-4. Demonstration of a Procedure Simulation, Screen 4.

The next procedure to be demonstrated is one in which the student must operate an item of equipment. The task is to set the fire control switch to emergency. As in the example of the hydraulic gauge, the student must first locate the fire control switch in the context of a gunner's station. After selecting the correct location, Figure 20-5 would appear on screen. The student must reposition the switch from the down position to the up position by placing the cursor on the switch and cycling the switch by pressing the left mouse button. When the switch is in the correct position, the student presses the right mouse button tc accept that position. The "Good" message then appears at the bottom of the screen.

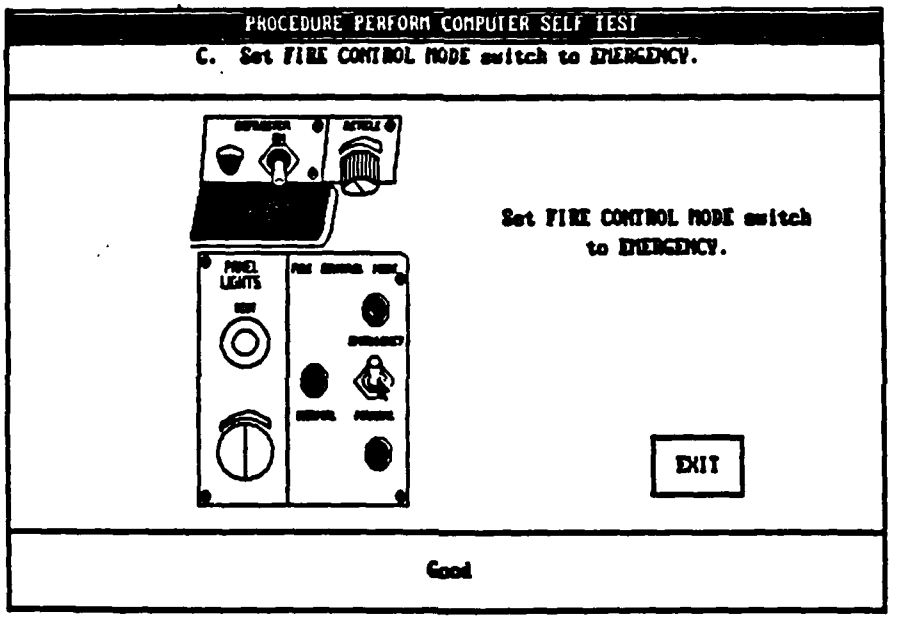


Figure 20-5. Demonstration of a Procedure Simulation, Screen 5.

After a three-second delay, a screen containing the information in Figure 20-6 appears, presenting the next step in the procedure. The task is to determine if main gun or turret drift is present. To make this determination, a close-up view of the gunner's primary sight is needed. To obtain this view, the student must locate the primary site, move the cursor to the correct location and press the left mouse button. A view of the sight appears as shown in Figure 20-7, except that the sight picture is capable of movement in the demonstration software. In this instance the student observes the sight and detects motion. The student then places the cursor on the box labeled "Drift, Notify Organizational Maintenance" and presses the left mouse button.

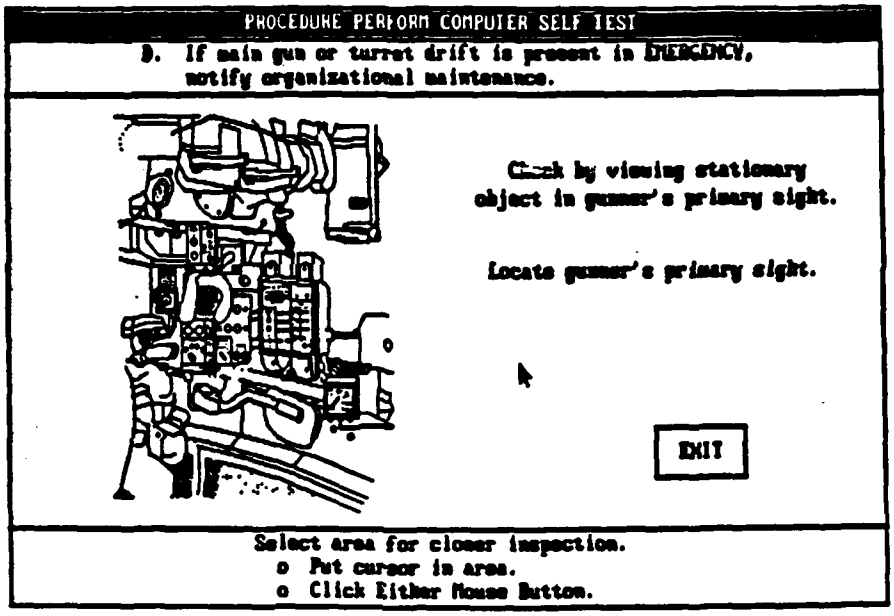


Figure 20-6. Demonstration of a Procedure Simulation, Screen 6.

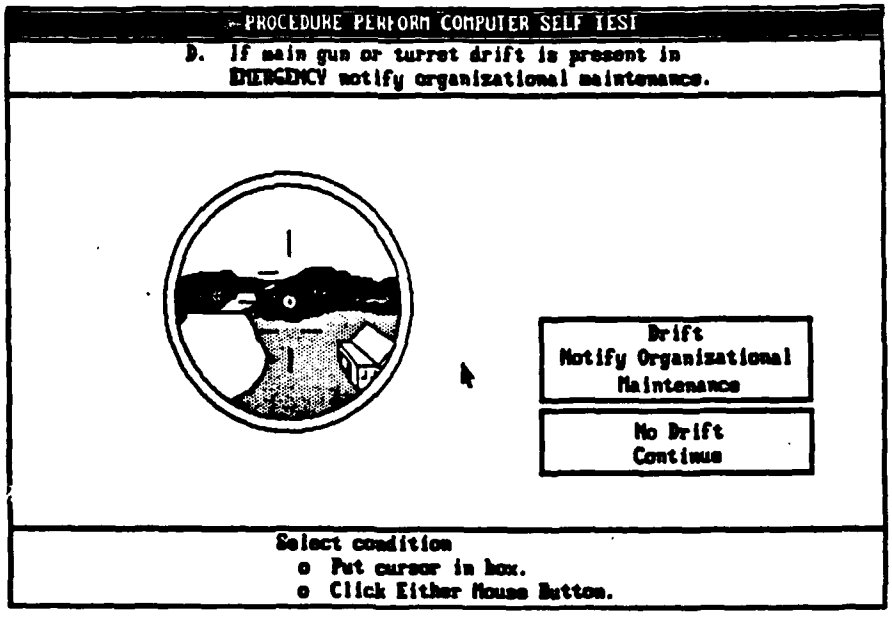


Figure 20-7. Demonstration of a Procedure Simulation, Screen 7.

In all cases where the status of an item of equipment indicates a problem (e.g., the reading on the hydraulic gauge is out of the appropriate range, main gun or turret drift is present), and the problem has been reported by the student, the following message appears: "To avoid terminating the exercise at this point, the last step is being reset. Try again." In the case of the hydraulic gauge, the new reading might be appropriate allowing the student to select the response "Oil Pressure Okay, Continue" and move on to the next screen. This feature of the program ensures that students will have the opportunity to observe items of equipment when no problems are indicated as well as when problems are indicated.

The next part of the demonstration illustrates how the student can practice correcting a problem situation using procedure simulation. The procedural step to be trained is to determine if drift exists and, if so, to adjust the system to eliminate the drift by turning the two Normal Mode Drift Knobs. Figure 20-8 illustrates a note screen which provides the student with an opportunity to practice using the mouse to manipulate the computer graphic knobs in question. This is a skill which is required to train with computer simulation but not required in the operational setting. When the student feels that he or she has mastered this activity, the student can select the continue option at the bottom of the screen to continue with the exercise. Figure 20-9 shows the screen which is used to practice the procedure of eliminating drift by adjusting Normal Mode Drift Knobs.

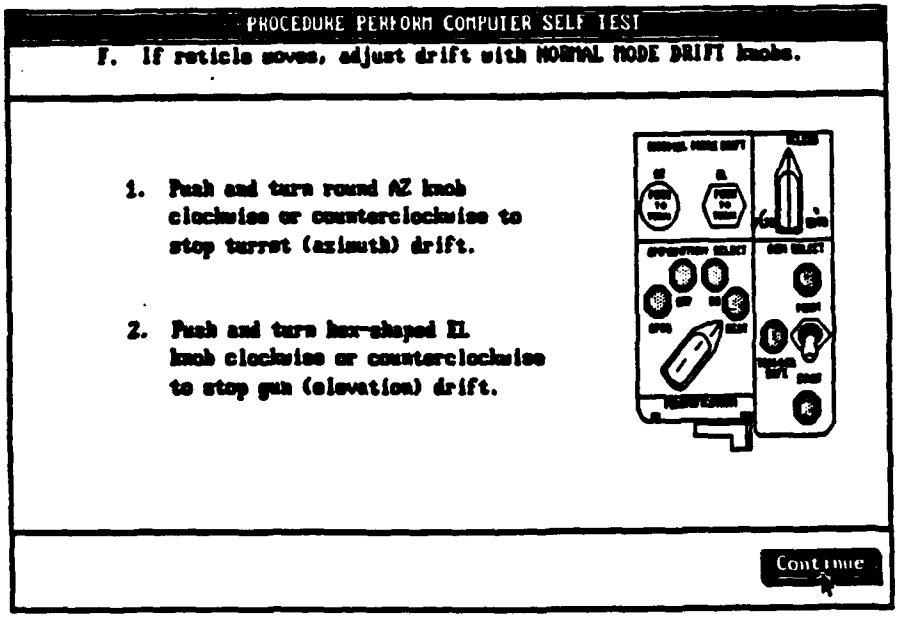


Figure 20-8. Demonstration of a Procedure Simulation, Screen 8.

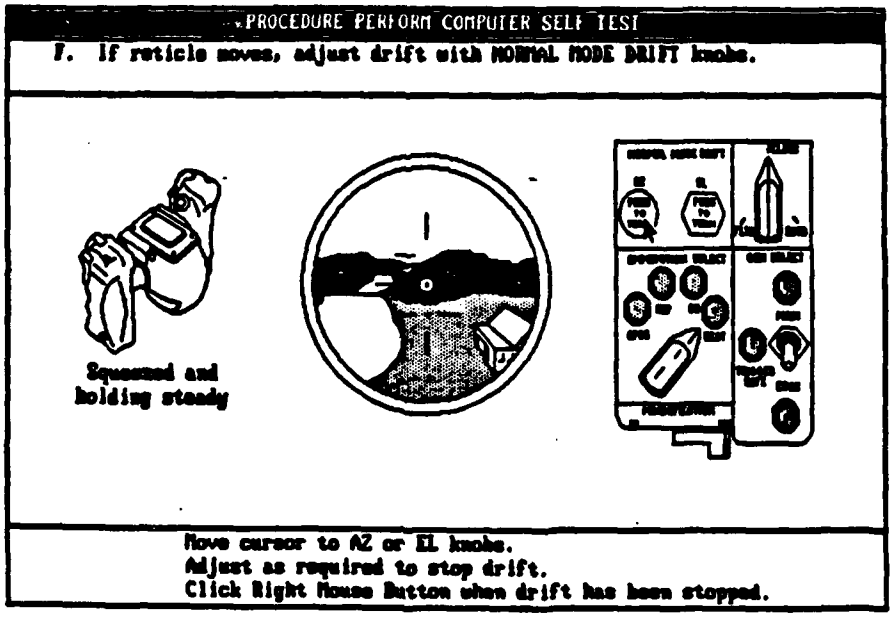


Figure 20-9. Demonstration of a Procedure Simulation, Screen 9.

Authoring Procedure Simulations with "Procedure Simulation Produced by Rehearsal (PROSPR)"

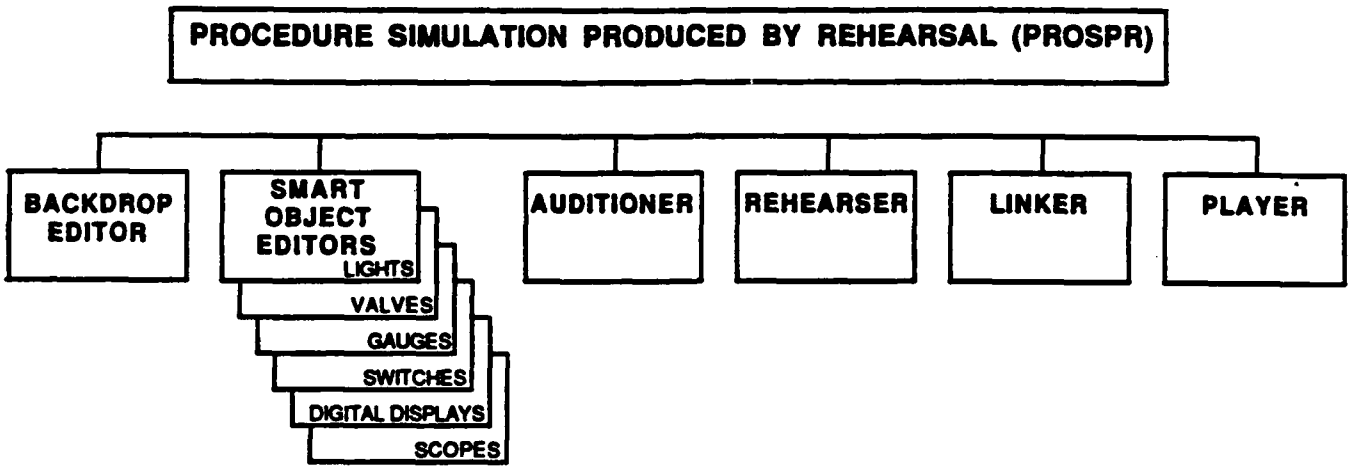
The authoring system (PROSPR) used to create the procedure simulation is a second generation system based on an original system developed for the NTIPP by Mears (1987). While entirely new code was created to carry out the authoring tasks, the basic approach was formulated in the earlier NTIPP work.

The emphasis was on smart screen design and display algorithms, not on authoring system development. However, a set of authoring tools was needed for use in creating the demonstration procedure simulation. The result was the development of a set of authoring routines that served the immediate purpose, and that also form the basic architecture of a special-purpose authoring system for procedure simulations. There was no attempt to make the tools user friendly, or to document the process of using the tools, or to create smart objects for the library other than those required for the demonstration, since this was outside the scope of the task and available resources. However the basic functions for a general purpose procedure simulation authoring system now exist and have been successfully used to create procedure simulations.

The project proponent, PM TRADE, and other Army organizations expressed an interest in developing the PROSPR demonstration software into a complete, user-friendly authoring and delivery system and requested information about the status of the software and/or copies of the software for the purpose of examination. Therefore, the government prepared a critique of the PROSPR authoring and delivery software and a user's guide for the authoring and information delivery portions of the demonstration software.

Six authoring system functions were built into PROSPR and are listed in Figure 21. Each of these functions are briefly described below. For a more detailed explanation of the authoring system, see Appendix E.

Backdrop Editor. This routine is used to create and save background objects. These are "dumb" graphic objects upon which smart objects will be placed. For instance, a drawing of the front panel of a piece of equipment can do none of the functions listed for smart objects, it is merely a static drawing. However, to make an Action Screen, a smart graphic of a switch can be placed on top of the dumb switch in the background graphic.



BACKDROP EDITOR: Used to Create and Save Background Objects.

SMART OBJECT EDITORS: Used to Create and Save Smart Objects Such as Lights and Switches.

AUDITIONER: Used to Select and Observe Individual Smart Objects in Action.

REHEARSER: Used to Assemble Procedure Simulation Screens.

LINKER: Used to Order the Screens of a Procedure.

PLAYER: Used to Place Linked Screens on the Display, Handle Student Input and Manage the Operation of the Interacting Smart Objects.

Figure 21. PROSPR Functions

Smart Object Editors. These editors are used to create graphics and embedded code for smart objects. Each class of objects has its own editor. For instance, the three-position switch editor was used to create smart graphics of switches which can be set to three different positions. The code defines the behavior of the switch, but does not limit the appearance of the switch. Therefore, a three-position switch might look like either of the switches shown in Figure 22, or it might look like any other switch which has three positions. The author simply creates graphics of the three positions in association with the generic code for that type of switch.

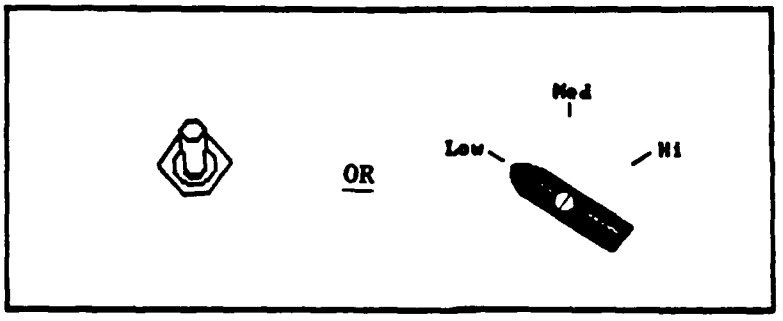


Figure 22. Three-Position Switches

Other behaviors automatically built into the switches include movement routines for switch positioning, memory of what other objects are affected by the switch position, messages available to other objects for critical switch positions, text messages to be displayed for a student if a student has not responded within specified time limits, and a variety of other functions peculiar to switches. Once the specific type of smart object editor exists, the author can easily and quickly make other objects of the class without programming skills. Smart objects are kept in a library so that they can be accumulated across projects and can be used in subsequent efforts.

Auditioner. With the auditioner, the author retrieves smart objects from the library and exercises these objects on the screen. After observing available objects, the author chooses the objects needed for a simulation and notes any changes to the appearance of the graphic objects needed to make those objects match the appearance of the operational objects. If appearance changes are required, the author can use the appropriate smart object editor to make the changes.

Rehearser. This routine is used to assemble the various smart and dumb objects into the dynamic screens used in the simulation. All dependencies and interactions among objects are established with this routine. For instance, if the procedure requires that when a power switch is turned on , the power light changes to red, then this relationship is established with the rehearser. The author also uses the rehearser to define the actions required of the student. In addition, the layout of the screen is determined by the author using the editors in this routine, except for certain layout decisions made automatically by the program.

Linker. The linker is used to sequence the screens to be used in the simulation. Many times of jumps are possible as the student proceeds through the simulation, due to choices in menus or randomly generated failures in the simulated system. Other jumps occur along a predetermined sequence of frames. Within the linker routine, the author is allowed to review individual screens. The linker records all decisions about screen ordering, whether these are made automatically by the program or manually by the author. When the author selects the save option, all the linking information is stored with each screen object.

Player. The player object uses the instructions held in each screen to manage the operation of the simulation by working like a stage manager, directing the presentation based on stored instructions and student input. Specifically, the player object:

- positions objects on the display in the correct location and correct state;
- manages input from the student;
- passes messages on to appropriate smart objects, insuring correct changes of state (e.g., student uses mouse to turn a switch on, then player instructs switch to change state, and then player instructs dependent light to assume changed state)
- manages timed events (e.g., helps a light to blink)
- checks the position of objects on appropriate student input

- skips screens that are not appropriate to the current mode
- withholds or presents information based on the current mode or student input
- routes the simulation to the correct screen when an "EXIT" area is selected
- manages random events
- manages feedback provided by text and flashing objects
- manages pop-up notes and menus
- places and removes prompts between steps
- manages the resetting of objects to their state at the beginning of the step when the student has performed incorrect manipulations

SECTION V:
RESULTS, CONCLUSIONS, AND RECOMMENDATIONS

The findings of this research should be of interest to individuals and organizations attempting to:

- apply object-oriented languages, especially Smalltalk/V, to the development of a visual programming language for use by nonprogrammers in creating procedure simulations;
- convert paper-based technical manuals to an electronic format;
- design screens for effectively conveying technical information in an electronic format;
- find low-cost methods for applying microcomputers to the conduct of technical training.

Results

Display algorithms were developed to retrieve text and graphic data elements from data files and display the data in different formats for technicians of differing skill levels. Depending upon information needs, the technician can review written descriptions of task steps, call up static graphics and animated graphics to clarify procedural text, call up locator graphics to help locate a particular item of equipment in the context of a larger system and/or call up detailed descriptions of task steps that are so complex that they warrant being addressed as a task in their own right.

Browse techniques were developed for use by technicians to scan a procedure and return to a previous position without the problem of becoming lost or disoriented. Options at the bottom of each screen allow users to "mark" a location in the procedure and/or store a series of frames for later review. "Maps" are provided for each procedure that allow users to quickly assess their location within a particular procedure.

A prototype authoring system for electronic technical manuals was developed for use by non-programmers. Hypertext and graphic scanning techniques are combined with the use of display algorithms that format technical information "on the fly." The user of the authoring system loads graphics in the database using an optical scanner and then prepares a "definition" for each procedure (i.e., lists procedural steps, indicates which graphic applies to each step and indicates cases where a procedural step is addressed by another entire procedure). These procedural "definitions" and graphics are then used to create frames as they are needed by the user. Use of this method for creating frames

greatly reduces the potential storage requirements, in comparison with attempting to store each unique frame.

Formats for "smart' display frames used in procedure simulations have been developed and can be used in procedure simulations supporting a variety of equipment. These frames include "locator screens" which can be used to train a technician to identify equipment components and "actions screens" which can be used to train a technician to check/operate equipment.

Using a visual programming technique, a prototype authoring system was created and demonstrated which allows a non-programmer to build a procedural simulation. The cost of developing such an authoring system with traditional program techniques is expected to be much greater than that of the cost of creating the present authoring and delivery systems.

The procedural simulation developed using the prototype authoring system ran on a low-cost IBM AT type computer. Such a computer has already been adopted for use in the Army's Electronic Information Delivery System (EIDS).

Software development was not an objective of this effort. However, information delivery and "rudimentary" authoring system software were developed for procedural job aids (PJAPS demonstration software) and procedural simulations (PROSPR demonstration software). These demonstration software packages were reviewed to assess their developmental status from the separate perspectives of users of technical manuals and authors of such manuals. These critiques are provided in Appendixes D and E. A brief summary of these critiques is provided below.

- Both software packages require improvements in certain menus employed by technicians and authors. These menus are created during the authoring process (e.g., lists of technical procedures, lists of graphics), and the current program limits the title of menu items to eight character MS-DOS file names. Menus with complete procedure and graphics titles would make the PJAPS much easier to use.

- Disk swapping is a mild problem in the information delivery and authoring applications of both software packages. However, it is important to note that the demonstration software packages address only a small portion of the technical information that would be addressed by, for example, a comprehensive electronic technical manual. Attempting to use the current software for full scale application might enhance the disk swapping problem and result in unacceptable delays for technicians, students and authors.

Conclusions

The screen formats for an electronic technical manual demonstrated in this effort go far beyond the expected functions of a simple electronic "page turner". The screen formats were designed to address the anticipated information needs of users varying in skill level. In effect, the end product is both a technical manual and a training device. Although the PJAPS technical manual does not provide performance feedback and the opportunity to practice, PROSPR software can be used in combination with PJAPS software to give technicians the opportunity to practice locating equipment components, checking the status of equipment and operating equipment. The combined use of PJAPS and PROSPR software offers a means of providing electronic technical manuals and low cost maintenance/operation training on new equipment. Such an application helps to meet the need for embedded training capabilities (Strassel, Dyer, Roth, Alderman and Finley, 1987).

The cost of transferring existing paper-based technical manuals into an electronic format is a critical problem, because the U.S. Army has a large number of paper-based technical manuals in its inventory. The technical procedures for the Cobra Helicopter, for example, are contained within approximately fifty-eight thousand pages divided among ninety-eight manuals. The cost of transferring technical manuals to the PJAPS format cannot be estimated at this time. The limited amount of content material from technical manuals addressed in the PJAPS demonstration software did not allow for developing comprehensive procedures necessary for authors to use in transferring the contents of technical manuals to the PJAPS format in a cost-effective manner. The task of transferring a particular manual to PJAPS requires a careful analysis of the contents of that manual to organize the contents in a manner which reaps the full benefits of electronic publication. Alternative formats which essentially employ microcomputers as electronic page turning devices for technical manuals might allow for less costly transfer of existing paper-based manuals to electronic manuals, but the end product would lack the training benefits offered by the PJAPS format. On the other hand, heuristic methods, combined with the application of data base management and parsing software might reduce the cost of transferring the large number of technical manuals in the Army's inventory to the PJAPS format.

Recommendations

The screen formats illustrated in this research, and the software created to implement these formats, have implications for a wide range of potential applications. These applications include converting paper-based technical manuals to electronic technical manuals, original development of technical manuals in an electronic format, the preparation of low cost procedure simulations, embedded training and console layout simulation.

A series of research and development objectives which provides input for all of these applications is presented below.

- Assess the effectiveness of improved PJAPS and PROSPR software
- Identify the major decisions to be made in transferring information from existing technical manuals to the PJAPS format, and identify commercially available or government-owned software which might be employed in a trial application of this process.
- Assess the extent to which PROSPR software can be used by non-programmers to develop procedure simulations and estimate the cost of producing these simulations.

REFERENCES

Chernoff, A. P., Joyce, R. P. , & Nauta, F. (1985). Maintenance job aids in the U.S. Navy: Present status and future directions. Orlando, FL: Naval Training Equipment Center.

Fuller, J. J., LeBeau, R. P., Mavor, A. S., Post, T. J., & Sawyer, C. R. (1987). Test and evaluation of the Navy Technical Information Presentation System (NTIPS): AN/SPA-25D field test results. Bethesda, MD: David W. Taylor Naval Ship Research and Development Center.

Fuller, J. J., Post, T. J., & Mavor, A. S. (1987). Test and evaluation of the Navy Technical Information Presentation System (NTIPS): F-14A field test results. Bethesda, MD: David W. Taylor Naval Ship Research and Development Center.

Nauta, F. (1985). Alleviating fleet maintenance problems through maintenance training and aiding research. Orlando, FL: Naval Training Equipment Center.

Post, T.J., & Price, H.E. (1973). Development of optimum performance aids for troubleshooting. Falls Church, VA: BioTechnology, Inc.

Schurman, D. L. & Kincaid, J. P. (1987). Job-site evaluation of the Personal Electronic Aid for Maintenance (PEAM). (in press) Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Strasel, H.C., Dyer, F.N., Roth, J.T., Alderman, I.N. and Finley, D.L. (1987). Implementing embedded training (ET): Volume 2 of 10: Embedded training as a system alternative. (ARI Research Product 88-22) Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Thomas, J.N., Braby, R., & Mears, J.E. (1988). Format model for computer-aided procedure training materials: two field studies at the Trident Training Facility, Bangor (NTSC Technical Report 88-044). Orlando, FL: Naval Training Systems Center.

APPENDIX A
INITIAL AND ADVANCED PRACTICE FLOW DIAGRAMS

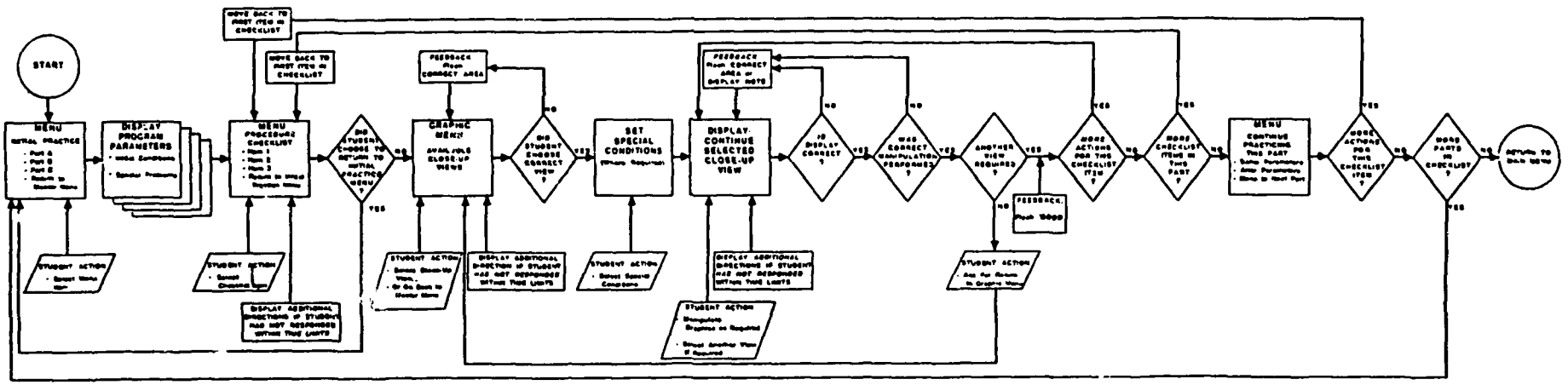


Figure A-1. Initial Practice Flow Diagram

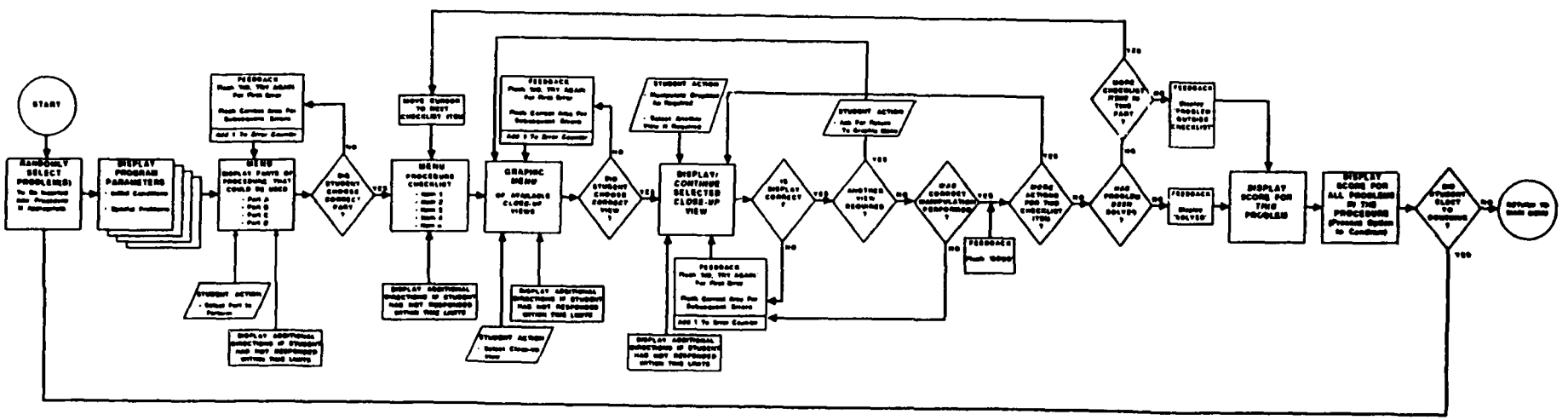


Figure A-2. Advanced Practice Flow Diagram

APPENDIX B
DESIGN OF THE PROSPR

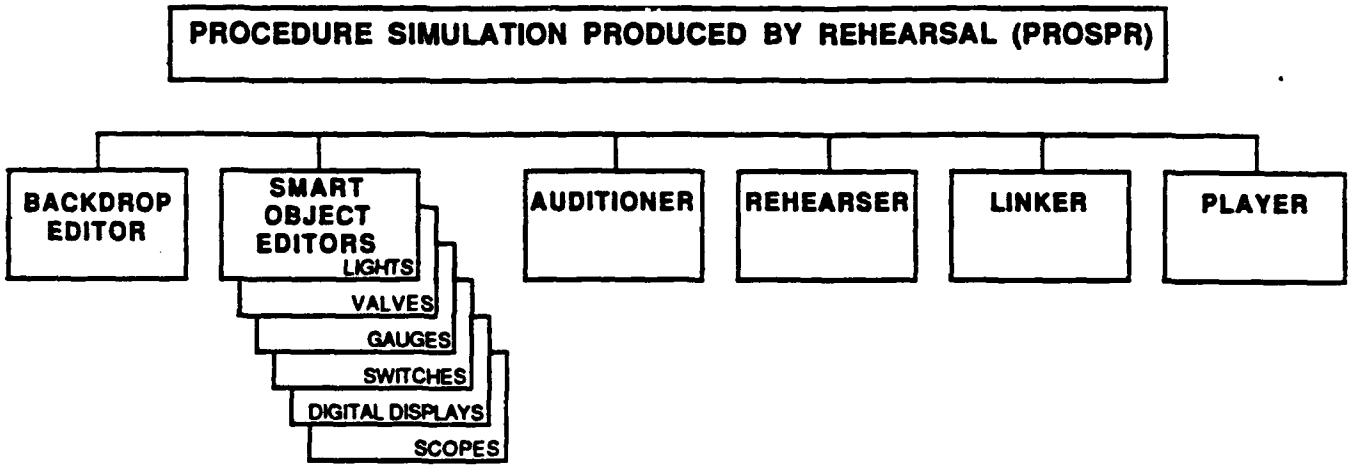
The development of the Procedure Simulation Produced by Rehearsal (PROSPR) was begun by defining the procedural objects and types of simple actions required in procedure simulation courseware. All the objects should have an exterior view (e.g., a valve "looks" like a valve) and should act like the object they represent. The author never needs to know exactly how an object knows what to do, only that the object knows how to respond to messages appropriate for that object. For example, since instances of lights are desired, a class called "Light" was included. Since an author might want a light to be on, off, or blinking, methods were included with the Light class instructing specific instances of that class (i.e., any specific light) how to turn off, turn on or blink. The author never sees the Smalltalk/V code of the action methods. but by sending a light object, the message "on", "off" or "blink", the light can carry out the desired action.

The author is allowed to choose from a dictionary of existing lights, or create one of his/her own using a light editor. Likewise, appropriate actions for valves, switches, gages and digital displays (and editors for the creation of these objects), are part of the PROSPR. By defining the basic classes of objects needed for procedure simulation and defining the basic methods for each class of objects, the basis for the PROSPR was developed.

After the basic procedural objects were defined, the objects which define the environment and relationship within a procedure had to be created. Figure B-1 presents an overview of the PROSPR, showing the basic purpose of each major subsystem.

An author should be allowed to use a graphics editor to create an overview or backdrop object. The FreeDrawing graphic editor class exists in Smalltalk/V but it lacks many features essential to create and save the PROSPR backdrops. By creating a Backdrop Editor Class as a Subclass of the FreeDrawing class and inheriting its desirable features, it was necessary only to implement those missing methods to make the Backdrop Editor Class fully functional. While backdrop objects do not know how to "act". they serve to give courseware screens a more realistic appearance of the piece of equipment depicted. The Backdrop editor provides the means for an author to easily create and save these background graphics.

The Backdrop Editor contains many useful features for creating and saving Backdrops. Besides the typical commands of most drawing programs (lines, circle, rectangle, ellipse, fill, etc.), the Backdrop editor has a Zoom capability to allow simple editing at the pixel level. The backdrop editor also contains the ability to import pictures created by scanning line art from



BACKDROP EDITOR: Used to Create and Save Background Objects.

SMART OBJECT EDITORS: Used to Create and Save Smart Objects Such as Lights and Switches.

AUDITIONER: Used to Select and Observe Individual Smart Objects in Action.

REHEARSER: Used to Assemble Procedure Simulation Screens.

LINKER: Used to Order the Screens of a Procedure.

PLAYER: Used to Place Linked Screens on the Display, Handle Student Input and Manage the Operation of the Interacting Smart Objects.

Figure B-1. PROSPR Overview and Functions

existing paper documents. By bringing the scanned image into the Backdrop Editor environment, an author can easily edit the drawing to save for use in his procedure.

The PROSPR has an Auditioner object, a forum for authors to select the light, switch, valve, gage, and digital display objects and to "audition" these objects for possible selection in a procedure script. Using the auditioner, the author can place the selected object on the screen and see how the object responds to its group of appropriate messages. If existing objects are unsuitable, PROSPR allows an author to create an object of his/her choosing, to be added to the PROSPR object dictionaries for an author's future use.

The PROSPR has a Rehearser, an object used to assemble courseware screens. Concepts drawn from existing procedure courseware were used to design the Rehearser. The Rehearser was designed with the capability to generate standard types of screen objects and a procedure object, designed to hold these screen objectives collectively.

When the Rehearser is initially invoked, the system asks the user if he or she wants to create a new procedure or modify an existing procedure. If the user wants to create an new procedure, the Rehearser sends a message to the Procedure class requesting a new procedure object. The procedure object is composed of a collection of dictionaries to hold "smart objects" (e.g., lights, switches), backdrops, and screens. If the user wants to modify an existing procedure, the user then selects and instance from the system's procedure dictionary. After the procedure has been selected, the user can create or modify the screens for that particular procedure.

Creation of text screens by the Rehearser is a very simple task. Through menu selection, the author is allowed to place Backdrop or text objects anywhere on the screen. Any object on the screen can be moved or removed, and the screen object can be saved through menu selection. To save a text screen, the Rehearser sends a message to the Textscreen class asking for a new instance, installs all the author-defined features and a unique identifier,, and places the object in the procedure object's screen dictionary. By similar means, this screen can be later modified and saved.

The Rehearser enables menu screens to have all the features of a text screen. The menu screen must also have a menu object with at least two entries. By selecting from the Rehearser's menus and pointing the cursor, the author can place the menu screen's menu in any location. The author can also insert, delete, or modify the screen's menu entries, or view the created so far. when the author instructs the Rehearser to save the menu screen, the rehearser sends a message to the MenuScreen class

requesting a new menu screen object. This new object is initialized and stored in the procedure object's screen dictionary. Training Strategy Screens and Mode Screens are treated like menu screens, except that their menu each have only two entries, which cannot be deleted.

The Rehearser uses a three phase process for the creation of Note, Locator and Action Screens. In the Placement Phase, the author can select backdrops and/or specific smart objects to place on the screen. Any object can be moved or removed, and fixed regions (like visible exit boxes or invisible locator rectangles for Backdrops) can be placed on the screen. The author can only place text in the step pane. Finally, the author is allowed to select the sizes of the screen's step and message panes in this phase.

When the author has completed the task of placing objects on the screen, he or she exist the Placement Phase and starts the Dependency Phase. In this next phase, the author defines existing dependencies between objects on the screen. The author must select an independent object (the object causing the action) and the dependent object (the object responding to the action) by moving the cursor to these objects on the screen. For example, if throwing a toggle switch causes a light to come on, the switch would be the independent object, and the light would be the dependent object. The Rehearser "knows" the type of objects selected, and queries the author about the specific dependent actions. These dependencies are stored within each independent object's dependency directory for use when screen actions occur.

The author exits the Dependency Phase when all dependencies have been define and starts the Rehearsal Phase. In this phase, the author rehearses the screen, and the Rehearser "watches" the author's cursor movement to define input. Correct input in these screens can be of three forms; selecting an area, selecting an object, or manipulating an object. If the author wants the student to select an area or select an object, the Rehearser "watches" the author move the mouse and click on the object of correct response. When manipulating an object, the author is instructed to point to the desired object of student action. After selecting the object, the object is cycled through all its possible positions by the author repeatedly pressing the left mouse button. When the desired position is reached, the author presses the right mouse button.

When the author has completed a screen, he or she can view the action as the Rehearser displays the action sequence. At any time, the author can return to the Placement or Dependency Phases to make revisions in a slide. When the "save screen" option is selected, the Rehearser sends a message to the appropriate screen class requesting a new instance. This new screen is initialized and saved in the procedure's screen dictionary.

The Linker object takes the screen objects created by the Rehearser, represents their identifiers in iconic form, and allows the author to link these screen objects together to form the presentation order of the procedure. The Linker allows the author to view any screen created by the Rehearser for the selected procedure. The Linker object holds all the information about screen ordering, and when the author selects the save option, all the ordering information is stored with each screen object.

The final object present in the PROSPR is the Procedure Player. This object uses the instructions held in each screen of a procedure object to perform the procedure. By following the links held in each screen object, the Procedure Player follows the directions present with each screen. This object serves as the student's main module, with each procedure serving as input to this object.

APPENDIX C
HUMAN FACTORS GUIDELINES

HUMAN FACTORS ASSESSMENT OF THE "PROCEDURE JOB AID PRODUCTION SYSTEM"

I. Introduction

A human factors assessment aids in obtaining:

a) Improved human performance as shown by increased speed, accuracy, and less energy expenditure and fatigue;

b) Less training and reduced training costs;

c) Improved motivation for use through minimizing the need for special skills and aptitudes;

d) Reduced loss of time as human errors are minimized; and

e) Improved convenience and acceptance by the user/operator.

The guidelines adopted are taken from the 1985 document NUREG/CR-4227 titled "Human Engineering Guidelines for the Evaluation and Assessment of Video Display Units" by W. E. Gilmore.

There are many variables that affect the design and assessment of Video Display Units (VDUs) and the workplace. The system is assessed by visual examination for legibility, readability, and comprehensibility. While we do not explicitly address hardware specific variables and workspace layout, our main focus of the discussion below is the user interface including frame design and interframe considerations as they relate to performance/job aids.

II. Human Engineering Considerations

A. Text format and content: What should be the content of the prose and how should be displayed.

Guidelines and Comments on Their Implementation

Guidelines	Comments
1. The textual content should be concrete rather than abstract.	The text is written in a cook-book style and is by no means abstract.
2. The textual content should lie within the scope of the user's existing knowledge.	The user is generally knowledgeable with the terminology and when necessary could opt for higher levels for more detailed text including overviews.
3. The textual content should be logically and systematically structured.	The text is highly structured and presented with introductory notes and sequence of instruction.
4. Consistent format should be maintained from one display to another.	A human-engineered format is adopted consistently for all the displays.
5. Prose should be displayed conventionally, in mixed upper and lower case.	Yes, mixed upper and lower case is used, since the lower case is quite legible.

6.	Displayed text should be left justified to maintain constant spacing between words, leaving right margins ragged if that is the result.	Yes.
7.	In textual material, words should be displayed intact wherever possible, with minimal breaking by hyphenation betweem lines.	Yes.
8.	Displayed paragraphs should be separated by at least one blank line.	Yes.
9.	In textual display, every sentence should end with a period.	Yes.
10.	In textual display, the main topic of each sentence should be placed near the beginning of the sentence.	Yes.
11.	In textual display, short, simple, concise sentences should be used.	Yes.
12.	When speed of display output for textual material is slower than the user's normal reading speed, an extra effort should be made to word the text concisely.	Yes, the display output is fast enough. Also, the text is worded concisely.

B. Messages: A message may be a prompt, a diagnostic message generated by an error condition, or an information or status message.

Guidelines and Comments on Their Implementation

Guidelines	Comments
1. The computer should be capable of providing two levels of detail.	The messages are short and, hence, two levels of detail are not provided.
2. Messages should be strictly factual and informative.	Yes. No effort is made to make computer appear "human."
3. Message dialogue should not be hostile to the user.	The messages are not hostile to the user.
4. Messages should be constructed using short, meaningful, and common words.	Yes.
5. The message should consider the prior knowledge of the user and the user's context.	Yes.
6. Sentences should be kept as simple in structure as possible.	Yes.
7. Messages should require no transformations, computing, or interpolation.	Yes.
8. Messages should be stated in the affirmative and preferably in the active voice.	Yes.

9.	Items to be remembered by the user should be placed at the beginning of the message.	There are no items to be remembered by the user. Only simple actions.
10.	Items to be recalled by the user should be placed at the end of the message.	There are not items to be called by the user. Only simple actions.
11.	Items of lesser importance should be placed in the middle of the message.	Yes.

C. Abbreviations

Guidelines and Comments on Their Implementation

Guidelines	Comments
1. Only standard and commonly accepted abbreviations should be used.	Yes. For example, psi stands for pounds per square inch.
2. Abbreviations should be short, meaningful, and distinct.	Yes.
3. The system should permit abbreviations of inputted commands.	Not relevant. There is no input of commands involved. Mouse is the only input device.
4. Whenever possible, experienced users should be provided with a set of abbreviations for frequently used commands.	Not relevant. Mouse is the only input device.
5. In text and labels, complete words should be used in preference to abbreviations.	Abbreviations are used whenever complete words do not add to the user's performance or preference.
6. Abbreviations should be consistent in form.	Yes.
7. If an abbreviation deviates from the consistent form, it should be specially marked whenever it is displayed.	Abbreviations are always used in consistent form.

8. A dictionary of abbreviations should be available for on-line user reference.	This is not found necessary, since users are familiar with the abbreviations, and there is no input involved.
9. Abbreviations and acronyms should not include punctuation.	Yes.

D. Error Statements: Useful and meaningful error statements are designed for error guidance, correction, and control of error.

Guidelines and Comments on Their Implementation

Guidelines	Comments
1. Operator inputs, responses, or actions which could significantly degrade computer system or plant performance should not be dependent on a single keystroke.	Yes. Usually a prompt follows such an instance to give a second chance to the user.
2. The computer system should contain prompting and structuring features by which an operator can request corrected information when an error is detected.	The system automatically suggests corrective action.
3. When the computer detects an entry error, an error message should be displayed to the user stating what is wrong and what can be done about it.	Yes.
4. Error messages should be worded as specifically as possible, based on computer analysis of data handling transactions.	Yes.
5. The wording of error messages should be appropriate to a user's task and level of knowledge.	Yes.

6.	When a control entry must be made from a small set of alternatives, those correct alternatives should be indicated in the error message displayed in response to a wrong entry.	Yes.
7.	Error messages should be brief and consistent while being informative.	Yes.
8.	Error messages should be stated in polite but neutral wording, without implications of blame to the user, without personalization of the computer, and without attempts at humor.	Yes.
9.	Following the output of simple error messages, the user should have the option of requesting more detailed explanation for errors.	The messages are simple, hence, this facility is not introduced.
10.	When multiple errors are detected in a combined entry, some indication should be given to the user, even though complete messages for all errors cannot be displayed together.	The errors are simple because of single input device, mouse.
11.	When an entry error is repeated, some noticeable change in the displayed error message should be provided.	The errors are simple. This facility is not provided.
12.	Error messages should be output after a user's entry has been completed.	Yes.

13.	An error message should be displayed approximately two seconds after the user entry in which the error is detected.	Yes.
14.	System documentation should include, as a supplement to on-line guidance, a listing and explanation of all error messages.	Yes.
15.	When an error has been detected in a data/command entry, the cursor should be automatically positioned at the point of the first error (data field or command word) in addition to the display of an error message.	Yes.
16.	Following error detection, users should be prompted to re-enter only the portion of a data/command entry that is not correct.	Yes.
17.	When a user entry can be recognized as doubtful, in terms of defined data/ command validation logic, a cautionary message should be displayed asking the user to confirm that entry.	Yes.
18.	In addition to a clear text error message, an error identification number (ID) should precede each message.	The errors are straightforward. No error numbers are designed.
19.	Error messages should always state or clearly imply at least a minimum of: o What error has been detected. o What corrective action to take.	Yes.

13.	An error message should be displayed approximately two seconds after the user entry in which the error is detected.	Yes.
14.	System documentation should include, as a supplement to on-line guidance, a listing and explanation of all error messages.	Yes.
15.	When an error has been detected in a data/command entry, the cursor should be automatically positioned at the point of the first error (data field or command word) in addition to the display of an error message.	Yes.
16.	Following error detection, users should be prompted to re-enter only the portion of a data/command entry that is not correct.	Yes.
17.	When a user entry can be recognized as doubtful, in terms of defined data/ command validation logic, a cautionary message should be displayed asking the user to confirm that entry.	Yes.
18.	In addition to a clear text error message, an error identification number (ID) should precede each message.	The errors are straightforward. No error numbers are designed.
19.	Error messages should always state or clearly imply at least a minimum of: o What error has been detected. o What corrective action to take.	Yes.

20.	If an error is detected in a group of stacked entries, the system should process correct commands until the error is displayed. A suitable error message should be presented, and no more inputs processed until the error is corrected. The page displayed should be the one requiring correction, and when the error is corrected, the system should continue processing any stacked inputs in order until done or until another error is detected.	Not relevant because of no stacked entries.

E. Instructions: Clear, accurate, and complete instructions are presented.

Guidelines and Comments on Their Implementation

	Guidelines	Comments
1.	Words in instructions should be meaningful to the user.	Yes.
2.	Short words should be used in instructions.	Yes.
3.	Active voice and the affirmative case should be used in instructions.	Yes.
4.	Instructions should be patterned.	Yes. Upper case is used to emphasize certain words.
5.	Sentences of instructions should be formatted into thought units (segments).	Yes.
6.	When appropriate, illustrations should be used as the primary vehicle for information rather than words.	Yes.
7.	Illustrations should be appropriate for the type of information it conveys.	Yes.
8.	Illustrations should be placed close to the corresponding text.	Yes.
9.	Wording on illustrations should be minimized.	Yes.

10.	Tables and graphs should be captioned.	Yes.
11.	Instructions should be well organized.	Yes. Titles, headings, and introduction are used to facilitate comprehension, learning, and retention of instructions.
12.	When instructions must be rapidly accessed, a table of contents and/or an index should be provided.	Yes. A "menu" of contents is provided.
13.	The literary style of a set of instructions should be appropriate to it intended use.	Yes. Instructions are designed (technically) for the intended user.
14.	Instructions should have a clearly stated beginning and a well-developed summary.	Yes.
15.	Paragraphs of text should be short and should contain a single idea.	Yes.
16.	Instructions should be simple.	Yes.
17.	Instructions should state important items more than once.	Yes.
18.	Instructions should contain only essential information.	Yes.
19.	The amount of detail should be appropriate to the experience of the user.	Yes. Three levels of detail are allowed to accommodate different levels of experience /expertise.

20.	Instructions should motivate the user.	Yes.
21.	The sequence of the instruction should follow the sequence of actions required.	Yes.
22.	Short sentences, flow diagrams, algorithms, lists, and tables are superior to prose.	Instructions are given in prose. But a flow diagram could be opted by the user.
23.	The main topic of the instruction should appear at the beginning of the sentence.	Yes.
24.	All instructions should be tested on naive users before being finalized.	Not relevant at this stage of system development.
25.	Many-step instructions should use a two-column format.	Instead, three-level instructions are designed.
26.	In a list of specifications for service or supply, more than a part number should be given.	Yes. More details are given.

F. Frame considerations: Uncluttered screens are designed. The screens are divided into windows to help the user visually perceive that the screen is structured in some logical format. Specific areas of the screen are reserved for invariant information such as commands, status messages and input fields consistently on all screens.

Guidelines and Comments on Their Implementation

Guidelines	Comments
1. An obvious starting point should be provided in the upper-left corner of the screen.	Yes.
2. Specific areas of the screen should be reserved for information such as commands, status messages, and input fields; those areas should be consistent on all screens.	Yes.
3. The screen should be symmetrically balanced.	Yes.
4. Instructions on how to use a screen or process information should precede the screen text or be at the top of the text.	Instructions are given in a consistent place.
5. Both the items on display and the displays themselves should be standardized.	Yes.
6. An invariant field, including the page title an alphanumeric designator, the time, and the date, should be placed at the top of each display page.	Yes, but time and date are not included as they are not relevant.

7.	The last four lines (at least) of each display page should be reserved for variant fields.	Yes.
8.	Procedures for user actions should be standardized.	Yes.
9.	Each display frame should have a unique identification (ID).	Yes.
10.	Every frame should have a title on a line by itself.	Yes.
11.	Every nonmenu frame should have an abbreviated reference to an operating manual and paragraph number (on same lines as the title).	No. Would be superfluous for this system.
12.	When an output frame contains more than one page, the notation "page of" should appear right justified on the last line.	Yes.
13.	If the full number of frame lines is not needed, the frame layout should be balanced and uncluttered.	Yes.
14.	The manner of presentation of identical data should be based on the way the operator will use that data.	Yes.
15.	Information of a critical nature should be located toward the center of the screen.	Yes.

16.	Status information should be displayed near the top-right corner of the screen.	No. Consistent bottom middle location allocated.
17.	Code information, sampling plans, and other information that the operator may need to call up should be located on one of the sides of the screen near the top.	Not directly relevant, but the top title line is incorporated.
18.	Location coding should be employed to reduce operator information search time.	Yes.

G. Interframe Considerations

Guidelines and Comments on Their Implementation

Guidelines	Comments
1. Whenever possible, all data relevant to the user's current transaction should be included in one display page (or "frame").	Yes.
2. When the requested data exceed the capacity of a single display frame, the user should be provided easy means to move back and forth among relevant displays either by paging or scrolling.	Yes, by paging.
3. When a list of numbered items exceeds one display page and must be paged/ scrolled for its continuation, items should be numbered continuously in relation to the first item in the first display and should indicate the present maximum location.	Yes, continuation is indicated.
4. When lists or tables are of variable length and may extend beyond the limits of a single display page, their continuation and ending should be explicitly noted on the display.	Yes.
5. When display output contains more than one page, the notation "page x of y" should appear on each display.	Yes.

6. When scrolling is used, a consistent orientation should be adopted in user-system interface (USI) design as to whether the data are conceived to move behind a fixed display window (commonly called "scrolling") or the display window is conceived to move over a fixed array of data (here called "windowing").	Scrolling is not used, hence not relevant.
7. In applications where a cursor is moved freely within a page of displayed data, "windowing" should be selected rather than "scrolling" as the conceptual basis of display movement.	Not applicable. The display is static when the cursor moves.
8. The parameters of roll/scroll functions should refer to the data being revieved, NOT to the window.	No roll/scroll functions. Not applicable.
9. When the user may be exposed to different systems adopting different usage, any reference to scroll functions should consistently use functional terms such as "forward" and "back" (or "next" and "previous") to refer to movement within a displayed data set rather than words implying spatial orientation (e.g., "up" and "down").	No scroll function. Not applicable.
10. When using a menu system, the user at all times should be provided access to the main menu.	Yes.

11.	Displays should indicate how to continue.	Yes.
12.	User-terminal interaction tasks that are repetitive, time consuming, or complex should be assigned dedicated functions.	Yes.
13.	Required or frequently used data elements should be included on the earliest screens in the application transaction.	Yes.
14.	Page design and content planning should minimize requirements for operator memory.	Yes.
15.	When pages are organized in a hierarchical fashion containing a number of different paths through the series, a visual audit trail of the choices should be available upon operator request.	Yes.
16.	Sectional coordinates should be used when large schematics must be panned or magnified.	Yes.
17.	The operator should have some capability for controlling the amount, format, and complexity of information (e.g., core dumps, program outputs, error messages) being displayed by the system.	Yes.

18.	If the message is a variable option list, common elements should maintain their physical relationship to other recurring elements.	Yes.
19.	A message should be available that provides explicit information to the user on how to move from one frame to another or how to select a different display.	Yes.
20.	The system should be designed to minimize the number of display levels required.	Yes.
21.	When the operator must step through multiple display levels, priority access should be provided to the more critical display levels.	Yes.
22.	When the operator must step through multiple display levels, he or she should be provided with information identifying the current position within the sequence of levels.	Yes.
23.	A similar display format should be used at each level of a multiple-level display.	Yes.
24.	When the operator is required to accurately comprehend previously learned items appearing with a new list, the list should be kept small (about four to six items).	Yes.

25.	**Frequently appearing/disappearing commands/subcommands should be placed in the same place on the screen.**	**Yes.**

H. Menu Design: Single-choice simple menus are designed.

Guidelines and Comments on Their Implementation

Guidelines	Comments
1. Menu selection should be used for tasks such as scheduling and monitoring that involve little entry of arbitrary data and where users may have relatively little training.	Yes.
2. Computer response should be fast.	Yes, but it is also hardware dependent.
3. Each menu display should require just one selection by the user.	Yes.
4. When menu selection is the primary means of sequence control, and especially if extensive lists of control options must be displayed, selection should be accomplished by direct pointing (e.g., by light pen).	Pointing is done using a mouse. It is efficient.
5. When menu selection is a secondary (occasional) means of control entry and/or only short option lists are needed, selection should be accomplished by keyed entry of corresponding codes or by other means such as programmed multifunction keys labeled in the display margin.	It is not a secondary means of control.

6.	When menu selection is accomplished by code, that code should be keyed into a standard command entry area (window) in a fixed location on all displays.	Not the case.
7.	Menu options should be worded so as to permit direct selection by pointing or by code entry, rather than worded as questions.	Mouse is used efficiently.
8.	When control entries will be selected from a discreet set of options, those options should be displayed at the time of selection.	Yes.
9.	If menu selection is used in conjunction with (as an alternative to) command language, displayed options should be worded in terms of recognized commands or command elements.	Not the case.
10.	If menu selections must be made by keyed codes, each code should be the initial letter (or letters) of the displayed option label rather than an arbitrary number.	Not the case.
11.	If letter codes are used in menu selection, then insofar as possible those codes should be used consistently in designating options at different steps in a transaction sequence.	Not the case.

12.	If menu options are included in a display intended also for data review and/or data entry (which is often a practical design approach), the labels for control entry should be located consistently in the display and should incorporate some consistent distinguishing feature to indicate their special function.	Not the case.
13.	Displayed menu options should be listed in a logical order; if no logical structure is apparent, then options should be displayed in order of their expected frequency of use, with the most frequent listed first.	Yes.
14.	Displayed menu lists should be formatted to indicate the hierarchic structure of logically related groups of options, rather than as an undifferentiated string of alternatives.	Yes.
15.	If menu options are grouped in logical sub-units, those groups should be displayed in order of their expected frequency of use.	Yes.
16.	If menu options are grouped in logical subunits, each group should be given a descriptive label that is distinctive in format from the option labels themselves.	Yes.

APPENDIX D
USER'S GUIDE FOR PROCEDURE JOB AID PRODUCTION SYSTEM (PJAPS) DEMONSTRATION SOFTWARE

CONTENTS

CONTENTS (Continued)

LIST OF TABLES

Page

LIST OF FIGURES

USER'S GUIDE FOR PROCEDURE JOB AID PRODUCTION SYSTEM (PJAPS) DEMONSTRATION SOFTWARE

I. Introduction to the Procedure Job Aid Production System (PJAPS)

Intended Benefits of PJAPS

The Procedure Job Aid Production System (PJAPS) was developed as part of an effort to identify methods for organizing technical information for electronic delivery (i.e., "paperless" manuals). Electronic technical manuals are expected to replace the Army's current system of paper-based technical manuals, and these new manuals will be employed on an Electronic Information Delivery System (EIDS) or a Militarized EIDS (MEIDS).

Major benefits of providing technical information in the formats described by PJAPS, from the perspective of the user of technical information, are listed below.

- The level of detail of technical information delivered by the system can be adjusted by the user to meet his/her specific needs. Users requiring more detailed information to understand a procedure have the option of static illustrations or even, in selected cases, animated illustrations to clarify a written description of a procedure.

- The user can browse through technical information and mark or store information to review in greater detail at a later point in time.

- The electronic manual would provide all of the information an operator needs to maintain/use the items of equipment addressed by the manual and remove the need for the user to reference other technical manuals.

The benefits, from the perspective of manual producers, of using the PJAPS to produce "paperless" technical manuals are listed and described below.

- PJAPS is able to create frames "on the fly" to reduce the overall data storage requirement associated with electronic publication. A key concern in making the shift to "paperless" manuals is the large number of frames required to present information (i. e., an estimated 15 to 20 frames might be required to provide the information contained on one paper page) and the work required to produce these frames. Rather than storing each frame individually, algorithms embedded in the PJAPS software creates frames

from database components as they are needed by the user of the technical manual.

- The work of preparing a technical manual is confined to describing the task steps involved in operating/ maintaining equipment and preparing static and animated graphics to illustrate task steps. Editorial chores required to organize this information in a format tailored to meet the information needs of diverse users of the manual are performed by the PJAPS software.

Scope of PJAPS

In general terms, PJAPS is a database management system specifically designed to support electronic delivery of information. Using PJAPS, a technical writer describes a job task in terms of specific information needs (task steps, static graphics, animated graphics and general information termed "notes"). Algorithms embedded in the PJAPS software provide the user of the technical manual with this information in a manner which allows information displayed to be tailored to the information needs of the user.

PJAPS also assists in editing graphics to be included in electronic manuals. Rough versions of graphics to be employed in a manual can be loaded in the PJAPS database using an optical character reader (OCR) and revised using editing tools embedded in the PJAPS software.

PJAPS demonstration software does not include provisions for integrating word-processing software, using a thesaurus or assessing the reading levels at which text is written. Integration of these other activities are, of course, long term objectives of the development of PJAPS software.

Purpose of PJAPS Demonstration Software

PJAPS demonstration software contains specific examples of technical information delivered in electronic formats. This software also contains a prototype authoring system which might be used to, for example, convert paper-based technical information to an electronic medium.

The sample sections of an electronic technical manual contained in the software will give you an opportunity to decide for yourself whether the formats meet the criteria provided in Table 1. Similarly, a first hand review of PJAPS authoring tools will help you to decide whether these tools are "user friendly" and employ cost-effective procedures to create the diversity of frames required to effectively employ the concept of a "paperless" manual.

TABLE 1. CRITERIA FOR EFFECTIVE ELECTRONIC DELIVERY OF TECHNICAL INFORMATION*

- Provide for the individual technician to adapt the display of the procedural steps to the level of detail needed to perform the task
- Display a single step, or no more than a small set of related steps on the screen at one time
- Design a system of displays that supports the technician in learning to perform skills (i.e, moving from a need for detailed types of procedural information to being able to perform the procedure with a minimum reference to technical information).
- Provide detailed graphics of procedural steps for those technicians needing this type of information
- Provide for an efficient browse capability to allow the technician to easily move through the data related to the procedure without getting lost.
- Provide for immediate and easy access to all the step sequences that make up a procedure, including those steps common to many procedures

* Criteria extracted from "Organization of Technical Information for Electronic Delivery" by Braby, Mears and Hebert (1988)

Organization of User's Manual

The second section of this manual describes hardware and software requirements for running the PJAPS software, and it provides instructions for loading the software on a hard disk. The third section will guide you through the process of using a "paperless" technical manual with examples of procedures taken from paper-based manuals addressing M-1 tank gunner's duties relevant to the task of operating a ballistic computer. This section will also point out specific drawbacks in the software which need to be corrected through future research and development.

The third section will guide you through the process of using the PJAPS as an authoring tool. This section will also point out specific problems with the user friendly aspects of the software which need to be addressed by future efforts.

II. Installation of PJAPS Software

Hardware/Software Requirements

- PROSPR software will run on an IBM or IBM compatible PC/XT or AT with a 20 megabyte hard disk drive. However, you should expect long delays when using any system below the AT level.
- The system using PROSPR software should have a VGA graphics adapter for best results.
- The use of PROSPR requires a Microsoft-compatible mouse.

Installing Software on a Hard Disk

PJAPS software is provided on eight floppy disks. The recommended procedures for loading PJAPS software are as follows.

- Make a directory called "ST2" by typing MD ST2
- Type CD ST2 to change to the "ST2" directory

DISKS 1-5

- To load disks 1 through 5, type RESTORE A: C: ST2*.* /F (NOTE: Be sure to leave a space between the last asterisk and the forward slash F). The system will then prompt you to load each disk in turn.

DISK 6

- While you are still in the ST2 Directory, create a subdirectory called "procs" (type MD Procs)
- Change to the "procs" subdirectory (type CD PROCS)
- Copy Disk 6 into the subdirectory (Type COPY A:*.*)

DISKS 7 AND 8

- Exit the "procs" subdirectory
- Create a subdirectory called "graphics" in the ST2 directory (type MD GRAPHICS)
- Change to the graphics directory (type CD GRAPHICS)
- Type RESTORE A: C: ST2\GRAPHICS\ *.* /F
- Load disks 7 and 8 in response to screen prompts
- Exit the ST2 directory

III. Examining Sample Contents of an Electronic Technical Manual

Initiate PJAPS

- When the "C" prompt is on your screen, type CD ST2 to change to the ST2 directory. Type "V" to initiate PJAPS.

- The first screen you will see is a project title screen. To get into the portion of PJAPS which contains the sample portions of a technical manual, follow the instructions below.

Selecting Sample Procedures for Examination

The M-1 gunner tasks "operate a ballistic computer" and "troubleshoot problems in a ballistic computer" have been divided into a series of subtasks termed "procedures". The contents of this portion of the demonstration are composed of a number of these procedures (e.g., "erect crosswind sensors). Examining two or three procedures should provide you with an adequate picture of what a PJAPS technical manual looks like.

- When the title screen appears, press the right mouse button for a menu. This menu will give you the option to "run a procedure", "create/modify a procedure", "create/modify a graphic" or "exit PJAPS"

- Move the icon to the option called "run procedure" and press the left mouse button. (As a general rule of thumb, pressing the right mouse button will display a menu and pressing the left mouse button will indicate a menu selection.)

- A menu containing a list of abbreviated procedure titles will appear. (NOTE: There is naturally a drawback with this menu in that the user cannot interpret the abbreviated titles/ file names. The next version of PJAPS should include a screen containing full task titles which "point" to the file names in a manner which is transparent to the user). For your convenience, a list of task titles is provided below.

  -Blamp1 Replace Indicator Lamp *
  -Blamp2 Replace Computer Control Panel Indicator Lamp *
  -Cpsftst2 Rerun Computer Self Test *
  -Cpslftst Operate Ballistic Computer (Perform Computer Self Test) *
  -Cwindcln Service Crosswind Sensor
  -Cwindup Erect Crosswind Sensor
  -Cwindsto Stow Crosswind Sensor
  -Enmanimp Enter Manual Inputs
  -Gcfail1 Troubleshoot Computer Failure *

-Gcfail2 Troubleshoot Cant Sensor Failure
-Gcfail3 Troubleshoot Crosswind Sensor Failure
-Gcfail4 Troubleshoot Lead (Azimuth Rate) Failure
-Gcfail5 Troubleshoot Elevation Rate Failure
-Gcfail7 Troubleshoot Data Link Failure
-Gcfail8 Troubleshoot Laser Rangefinder Failure

- Select any procedure and press the left mouse button. For demonstration purposes, it is recommended that you select one with an asterisk (i.e., procedures with an asterisk contain animated graphics).

- A few seconds to a minute might be required before the next screen appears. (NOTE: The amount of time required to load certain files is a problem to be addressed in future efforts. You should consider that an entire electronic technical manual will contain many more files than does the current demonstration, and delays due to disk swapping would be much greater without corrective action.)

- In the middle of the new screen, a menu with two options will appear; "NOTES" and "THE PROCEDURE ITSELF". See Figure 1. The first option, "NOTES", provides important information about the task other than procedural steps. The second option provides a step by step description of the procedure. The labels, boxes and arrows at the bottom of the screen are in what might be termed a "control box". The functions of these elements will be described as the need for each element arises.

Reviewing Notes

- Move the icon to the word "NOTES" and press the left mouse button. Move the icon to the down right arrow and press the left mouse button. The specific notes for this task will appear on your screen.

- To obtain more information about a particular note, move the icon to that note and press the left mouse button. If a down arrow appears at the bottom right of your screen, this means that a graphic is available which helps to clarify the note. Move the icon to the bottom of the screen and press the left mouse button. A graphic will then appear on your screen.

- To return to the listing of task notes, move the icon to the arrow pointing up and press the left mouse button.

Reviewing Procedural Steps

- Move the icon to the words "THE PROCEDURE ITSELF" and press the left mouse button.

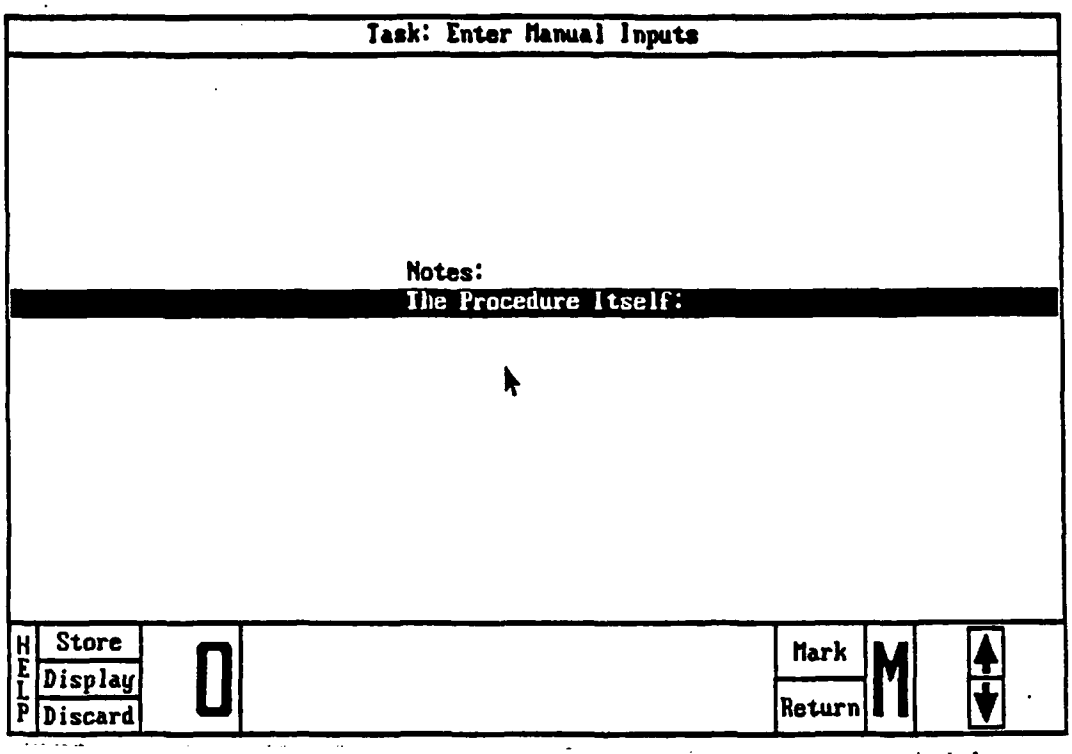


Figure 1. Screen Showing Format of Initial Menu within each PJAPS Procedure and Control Sections at Bottom of Screen.

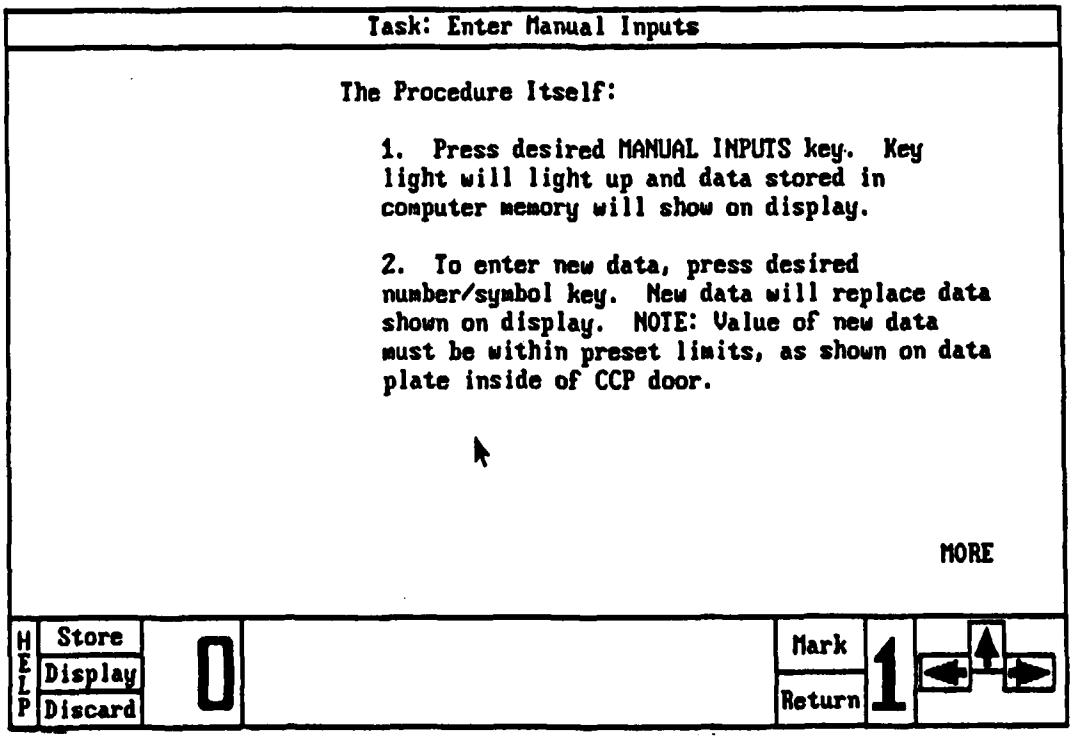


Figure 2. Example of a Procedural Steps Display within PJAPS.

- Move the icon to the down arrow at the right bottom of the screen and press the left mouse button. Procedural steps will then appear on the screen. See Figure 2. If the words "MORE" appear at the bottom right of the task steps, this means that there are more steps for this task covered on subsequent frames. Observe how each frame contains only a small amount of text to facilitate use of the guidance in a work setting.

- To go through each set of steps in the procedure, move the icon to the right arrow and press the left mouse button. Notice the number "1" in the box at the bottom right of your screen. This number indicates that you are at the first level of procedural detail.

Using Graphics to Obtain More Detailed Information About Procedural Steps

Static graphics. The least sophisticated form of graphic aid is the static graphic.

- Move the icon to the task step for which you want to obtain more detailed information, and press the left mouse button. If a down arrow appears, this means that a static graphic accompanies this procedure.

- Move the icon to the arrow pointing down and press the left mouse button. A figure will then appear on your screen along with a restatement of the procedural step. See Figure 3. Notice that the number in the box at the right bottom of your screen has changed to "2", indicating that you are at the second level of procedural detail.

Locator graphics. Locator graphics are designed to assist the user of a technical manual in locating the specific item of equipment within the context of a larger system (e.g., where is the ballistic computer on an M-1 tank?) This type of graphic is appropriate for aiding an individual during entry-level training.

- If information is required about the location of the piece of equipment shown in the figure, move the icon to the figure and press the left mouse button. A locator graph will appear which shows where the equipment is located in the context of the larger system. See Figure 4 for an example of a locator graphic.

- Locator graphics might not exist for all static graphics. In this case, the control box at the bottom of your screen will display the message "no overview exists for this picture".

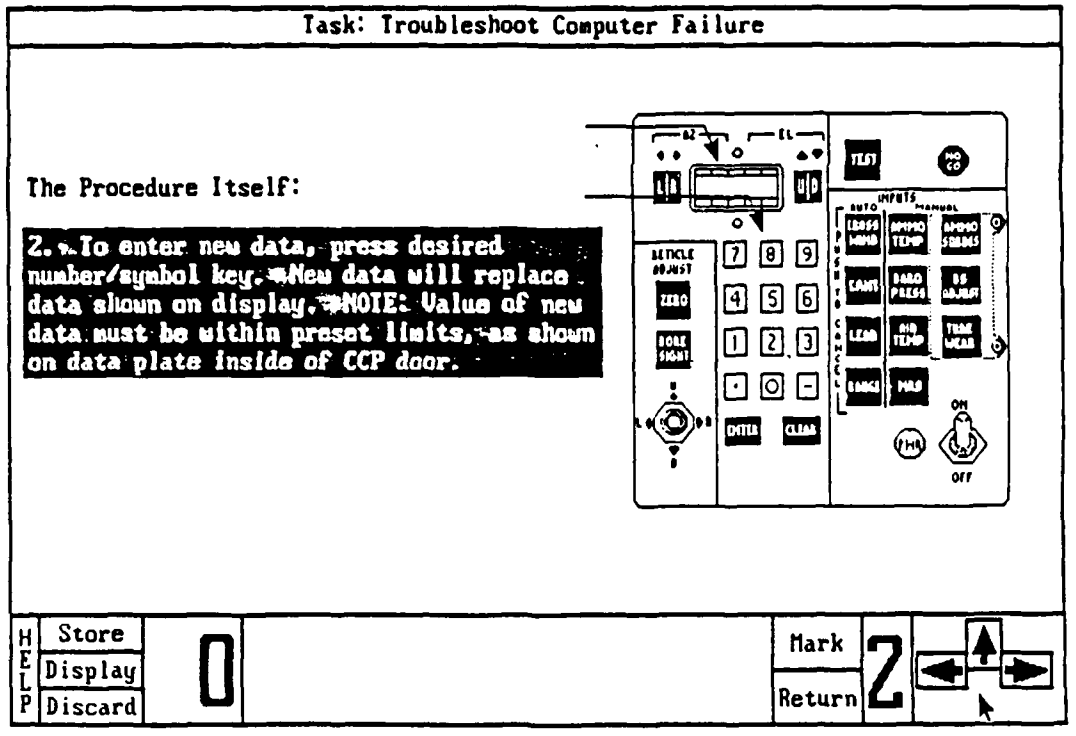


Figure 3. Sample PJAPS Frame Using a Static Graphic to Clarify a Procedural Step.

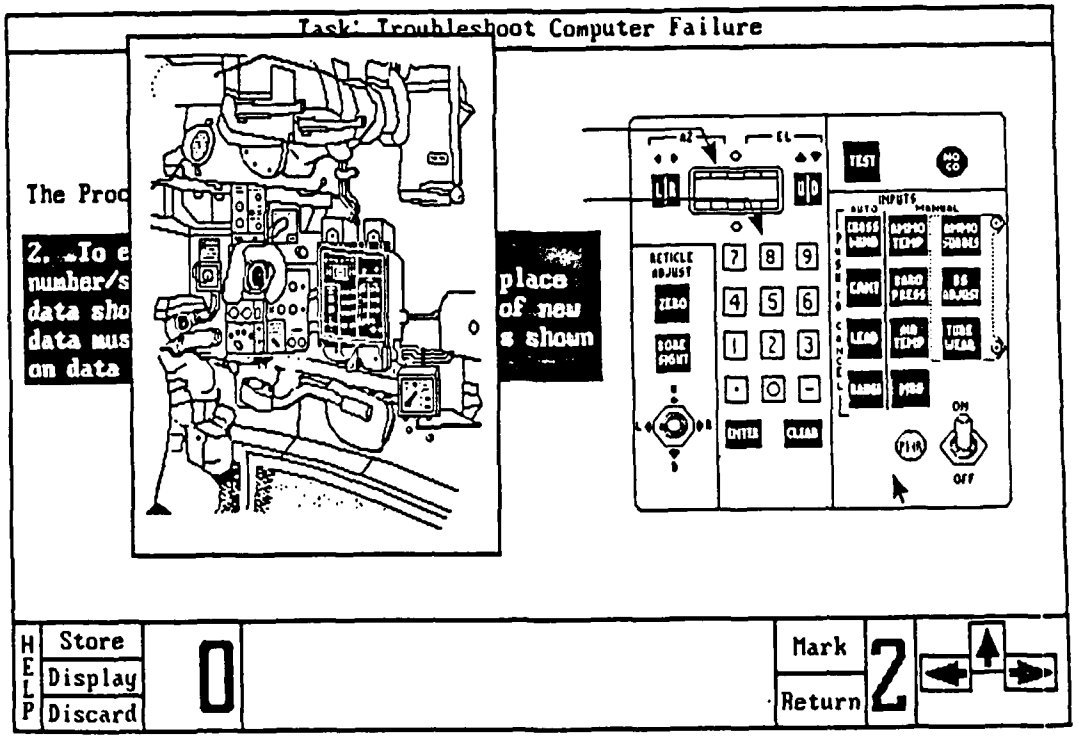


Figure 4. Sample PJAPS Frame Using a Locator Graphic to Help the PJAPS User Decide Where a Particular Item of Equipment is Located in the Context of a Larger System.

Animated graphics. For certain procedural steps there may be an observable change in an item of equipment after the step is performed. Animated graphics which show before/ after action status of equipment can help to clarify procedures.

- When you have a level 2 static figure on your screen, check to see if a down arrow is present in the bottom right corner of the screen. If so, an animated version of the figure is available.

- Move the icon to the down arrow and press the left mouse button for access to the animated graphic. The animated graphic will show a "before" and "after" screen in sequence. The first screen shows what the equipment looks like before the action described in the task step is taken, and the second screen illustrates the effects of the action on the piece of equipment.

Using Linkages to Gain more Detailed Information About Procedural Steps

- Certain of the procedures in the sample electronic technical manual show up as procedural steps within the context of other procedures. For example, the procedure "Rerun Computer Selftest" shows up as a procedural step within "Troubleshoot Crosswind Sensor Failure". If you already know how to rerun a computer selftest, then you would not need detailed information about this step. If you needed detailed information, PJAPS "linkages" would give you access to the entire "Rerun Computer Selftest" procedure, including all procedural steps and graphics.

- The value of PJAPS linkages can be illustrated by considering how paper-based technical manuals often handle the fact that one task can be a subtask within another task. In reading a technical manual, you will often be referred to another technical manual to obtain more detailed information about a procedural step. Further, when you check the manual to which you have been referred, that manual might refer you to yet another manual for more detailed information. PJAPS linkages bring all of these references to the user of the technical manual.

- You gain access to linkages in the same way that you gain access to the graphics which help to clarify procedural steps. Move the icon to the procedural step, press the left mouse button, move the icon to the down arrow, and press the left mouse button. It is important to note that you, as a user do not specifically ask for a linkage. By using the down area you are merely asking for more information. If a linkage exists, detailed procedural substeps will be displayed. If no linkage exists, static graphics will be displayed.

Browsing Through a Procedure

Procedural Maps. PJAPS was designed to provide the user with the ability to browse through the contents of an electronic manual and mark sections to be reviewed later in greater detail. One means of browsing through a PJAPS procedure is to employ a specific "browsing" function built into the PJAPS. This browsing function involves the use of a procedural map.

- Procedural maps are displayed by moving the icon to the screen level number and pressing the left mouse button. Figure 5 is a sample procedural map.

- The procedural map will have either three of four rows of blocks depending upon whether the procedure has any animated graphics. The fourth row represents animated graphics frames and the third row represents static graphic frames in the procedure.

- As you move the icon along the boxes in the third and fourth rows, note that the information box at the bottom center of the screen provides an abbreviated name/ description for each frame. When you reach one of interest, move the icon to the information box and press the left mouse button. The frame you selected will then appear on your screen. (To get rid of the procedural map without selecting a frame, move the icon to the bottom of the frame and press the left mouse button)

- Certain of the boxes in the procedural maps contain the letter "S". This means that the procedural step addressed by the box is a procedure in its own right within the electronic manual. That is, this step in the procedure is "linked" to another entire procedure. For example, any procedure which includes the step "Rerun Computer Selftest" will have a procedural map in which the box representing this step has an "S" in it, because "Rerun Computer Selftest" is a procedure in its own right.

Storing Frames. The storage function of PJAPS lets you store a number of frames within a procedure. In this way, you can set aside frames you feel deserve looking at in more detail (e.g., because they address task steps on which you need more information).

- When you see a frame that you want to store, move the icon to the "store" frame and press left mouse button

- When you are ready to look at the frames you have stored, move the icon to the "display" frame and press the left mouse button.

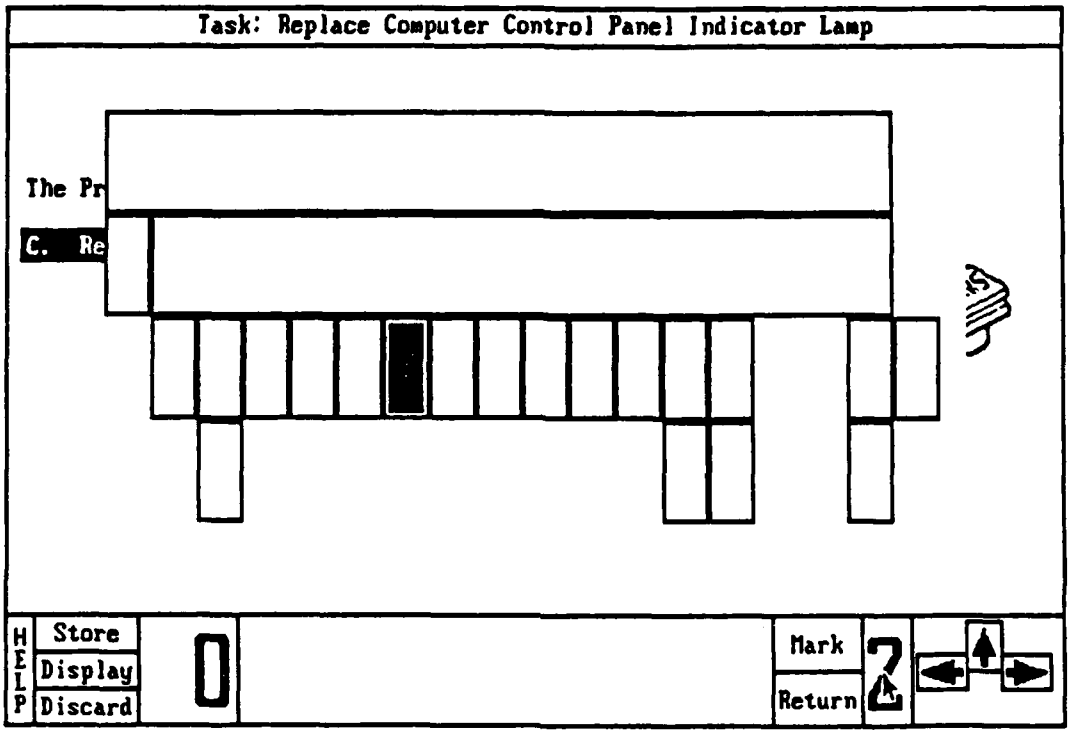


Figure 5. Sample PJAPS Procedural Map.

- To move forward or backwards in the sequence of frames you have saved, use the left/right arrows (Note: The order in which frames are stored is the order in which they are selected for storage)

- The frames are saved until you exit the procedure you are in or until you intentionally discard a frame. (NOTE: failure of the PJAPS to save stored frames when the user exits a procedure is a flaw to be addressed in the upcoming revision of the software).

- To discard a stored frame on your screen, move the icon to the "discard" box and press the left mouse button.

Marking Your Place. The PJAPS has a "page marking" function which allows you to mark your place in a procedure so that, if interrupted, you can quickly find your place.

- To mark a frame, move the icon to the box in the lower right portion of the screen labeled "mark" and press the left mouse button.

- To return to the marked frame, simply move the icon to the box labeled "return" and press the left mouse button

- (NOTE: There is a flaw in the mark function. If you exit a procedure, your mark will not be retained in memory. This problem needs to be addressed in future refinements of the software)

Making Hard Copies of Screens

- Making hard copies of most types of PJAPS screens is, admittedly, a slow/cumbersome process. Further, copies of most screens can only be printed using a LaserJet printer.

- When you want to make a hard copy of a particular screen, it is necessary to Press the "F2" key.

- Approximately 10 minutes are required for the system to begin printing after the "F2" command

Exiting a Procedure

- Move the icon to the top arrow and press the left mouse button to move the screen up to less detailed information. A menu will appear which gives you the option of exiting the procedure.

IV. Examining the PJAPS Authoring Process

Overview of Authoring Process

The PJAPS authoring process is divided into two major parts. One part is concerned with defining each procedure in terms of notes, procedural steps, graphics and linkages with other procedures. The outcome of this effort is a list of notes and procedural steps with a notation of the titles of the graphics associated with each note/procedural step. In cases were a procedural step is addressed by another entire procedure, the other procedure is listed as a linkage for the procedural step. (See Table 2 for an example).

The second part of PJAPS is concerned with the preparation of graphics. PJAPS graphics tools can be used to prepare original graphics or to edit a drawing which has been placed in the database through the application of an Optical Character Reader (OCR).

Getting Into the PJAPS Authoring System

- Type "CD ST2" and then "V" to initiate PJAPS

- When title screen appears, press the right mouse button for a menu. This menu will give you the option to "run a procedure", "create/modify a procedure", "create/ modify a graphic" or "exit PJAPS"

Modifying a Procedure

Procedure Modification/Creation Menu. Essentially the same authoring tools are used for creating or modifying a procedure. For reasons of simplification, the process of modifying a procedure will be discussed first.

- Select "create/modify procedure". A second menu will then give you the options "modify a procedure" or "create a procedure".

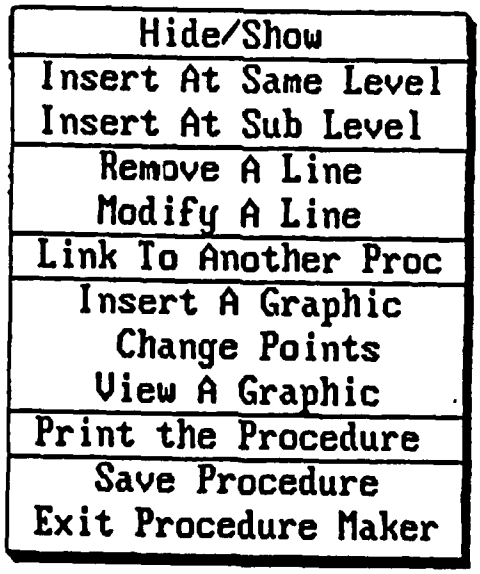
- Select "modify procedure". A menu will appear with the abbreviated names for the various procedures. Select a procedure.

- Next you will see a screen with the task name at the top and the phrases "notes" and "the procedure itself" immediately beneath. Press the right mouse button for a menu. The menu for modifying a procedure is shown in Table 3.

TABLE 2. SAMPLE PJAPS PROCEDURE PRINTOUT

```
Task: Troubleshoot Computer Failure
   The Procedure Itself:
      1.  Enter AMMO TEMP number into computer.
         LINK ***(enmaninp)*** LINK
      2.  Rerun computer self test.
         LINK ***(cpsftst2)*** LINK
      3.. Check computer display and NO GO light.
         GRAPHIC ***(gcslftst)*** GRAPHIC
      3a. If the word 'PASS' appears on the display, computer self test
      is completed.
         GRAPHIC ***(sftstk3)*** GRAPHIC
      3b. If NO GO light lights, go to step 4.
         GRAPHIC ***(gcnogo)*** GRAPHIC
      4.  Re-enter all manual inputs into computer.
         LINK ***(enmaninp)*** LINK
      5.  Recharge computer electronics unit battery.  Set TURRET POWER
      switch to ON for one hour.
         GRAPHIC ***(trtpwrsw)*** GRAPHIC
            Set TURRET POWER switch to ON.  Leave it set to ON for one
            hour.  This will recharge the computer electronics unit
            battery.
      6.  Rerun computer self test.
         LINK ***(cpsftst2)*** LINK
      6a. If the word 'PASS' appears on display, continue operation.
         GRAPHIC ***(sftstk3)*** GRAPHIC
      6b. If NO GO light lights, notify organizational maintenance.
         GRAPHIC ***(gcnogo)*** GRAPHIC
      -- End of Procedure --
```

TABLE 3. PJAPS PROCEDURE EDITOR MENU



Printing a Hard Copy of a Procedure. Obtaining a hard copy of a procedure before you start to modify the procedure is important because PJAPS usually does not display the entire text of a procedure on the screen. (NOTE: For procedural steps with lengthy text, PJAPS cuts off the last portion of the text when it displayed on screen. This is a problem which needs to be addressed by future revisions of the software).

- Select "Print the procedure" to obtain a hard copy of a procedure. An example of such a hard copy is provided as Figure 6. Note that a procedure is described in terms of the components listed below.

  - A statement of task steps
  - A notation of the graphics associated with each task step
  - A notation of linkages

Displaying the Procedure on Your Screen/Use. The "hide/show" function is used to control whether the procedure appears on screen, and it is used to decide how much of the procedure appears on screen.

- Move the icon to "the procedure itself" and press the left mouse button. Then press the right mouse button for a menu

- Move the icon to the menu item "hide/show" and press the left mouse button. The task steps, graphics and linkages will appear on your screen.

- Notice that the right hand portion of many procedural steps are cut off in the display.

- After the procedure is displayed, you can use the "hide/show " menu option to control the display of lists of graphics and linkages for individual steps. To use the "hide/show" menu option in this way, select a procedural step, call up the menu and select "hide/show". If the graphics/linkages for the steps are displayed before you select "hide/show", they will disappear from your screen. If the graphics/linkages are not displayed before you select "hide/show", they will appear in the display after you select "hide/show".

Deleting a Procedural Step, Graphic or Linkage. The system will not allow you to delete a procedural step if the task step contains any graphics or linkages. To remove a step, you must first delete the graphics and/or linkages. If you try to remove a step before graphics/linkages have been deleted, you will receive the message which follows "Cannot remove yet-Has substeps!"

- Move the icon to the step, graphic or linkage to be deleted, and press the left mouse button.

- Press the right mouse button for a menu and select the menu option "remove a line". The deleted line will disappear from your screen.

Modifying a Procedural Step. This method is appropriate whenever you wish to rewrite an exiting procedural step.

- Move the icon to the step to be rewritten and press the left mouse button.

- Press the right mouse button for a menu and select the option "modify a line"

- The screen will then prompt you to type the new/modified line

- When you finish typing, press "enter". The display will then be updated to reflect your changes

Adding a Procedural Step. This method is appropriate whenever you wish to add a procedural step where one has not previously existed.

- Move the icon to the step which immediately precedes the step you wish to add and press the left mouse button

- Press the right mouse button for the menu and select the "insert at same level" option

- The screen will then prompt you to type the line to be added. When you have finished typing, press "enter" The new step will then be displayed on your screen.

Adding a Linkage. In cases where a procedural step addresses a task/subtask covered elsewhere by an entire procedure, the linkage is used to identify this separate procedure.

- Highlight the step to which the linkage is to be added, press the right mouse button for the menu and select the option "link to another procedure".

- The screen will then display a menu of the procedures within the database. Move the icon to the title of the procedure to which the step is to be linked and press the left mouse button. The linkage will then be added to the procedure on your screen.

Adding a Graphic. This method allows you to add static or animated graphics to procedural steps using graphics already contained within the database.

- Move the icon to the step to which a graphic is to be added and press the left mouse button.

- Press the right mouse button for a menu and select the option "insert a graphic". A new menu will then appear with the options "insert a static graphic" and "insert an animated graphic". Select which ever option is appropriate.

- A third menu will then appear with the titles of the static or animated graphics in the database, as appropriate. Select the graphic you wish to include. A box will then appear at the bottom of your screen with the icon at the upper left hand corner. Press the left mouse button and the graphic you have selected will appear in the box, allowing you to make sure you have called up the graphic you wanted.

Exiting the Procedure Menu. Before you exit you must remember to save the changes you have made in the procedure. Call up the menu and select the option "save procedure". Call up the menu again and select "exit procedure editor".

Deleting an Entire Procedure. You cannot delete a procedure while you are using PJAPS. To delete a procedure, you must delete the file which contains that procedure. All such files will be contained in the "procs" subdirectory of the "st2" directory that you created when installing the PJAPS software. If you created a procedure called, for example, "Load", this procedure would be stored as the file named "load.prc" (i.e, all procedures are automatically given the file extension "prc").

- Type CD ST2\PROCS to enter the "procs" subdirectory where the files are stored

- Type DEL and the name of the file to be deleted (e.g., DEL LOAD.PRC).

- Unfortunately, deleting a procedure deletes certain references to the files but not others. Deleting a procedure automatically removes that procedure from menus of procedures (i.e., menus of procedures that can be run or modified). Deleting a procedure does not remove any cases where the deleted procedure has been listed as a linkage for another task. To delete these references you would have to check each procedure to which the deleted procedure might be linked and delete the linkage. (NOTE: The awkward process of deleting linkages after a procedure has been deleted needs to be investigated in future research. However, the cost of a "fix" to this problem may be too great relative to the benefits. The case may be that very few procedures are ever entirely deleted).

Creating a Procedure

- The guidance for creating a procedure is the same as that for modifying an existing procedure with the following difference. Slightly different methods are used to write the first step in a procedure. After the first step is written, the methods for creating/revising a procedure are identical.

- To begin writing a procedure, move the icon to "notes" or "the procedure itself" (depending upon which section you want to work on) and press the left mouse button. Press the right mouse button for the menu and then select the option "Insert at sublevel".

- Type in the first step of the procedure when prompted by the screen.

- To add steps beyond the first, use the same procedure you would use to add procedural steps (i.e., select the last procedural step and then call up the menu and select the option "Insert at Same Level").

- To add linkages and graphics use the same procedure you use in adding linking/graphics to an existing procedure (i.e., select the procedural step you want to work on and then call up the menu and select "Insert at sublevel").

- When you have finished the procedure, be sure and select the "save procedure" option from the menu.

- When you select the "exit procedure editor" option you will exit to the "procs" subdirectory where all of the procedures are stored. Type DIR and the name of the procedure you have created (with the extension ".prc" to call up the directory, and make sure your procedure has been saved.

V. Examining the PJAPS Graphics Editing Process

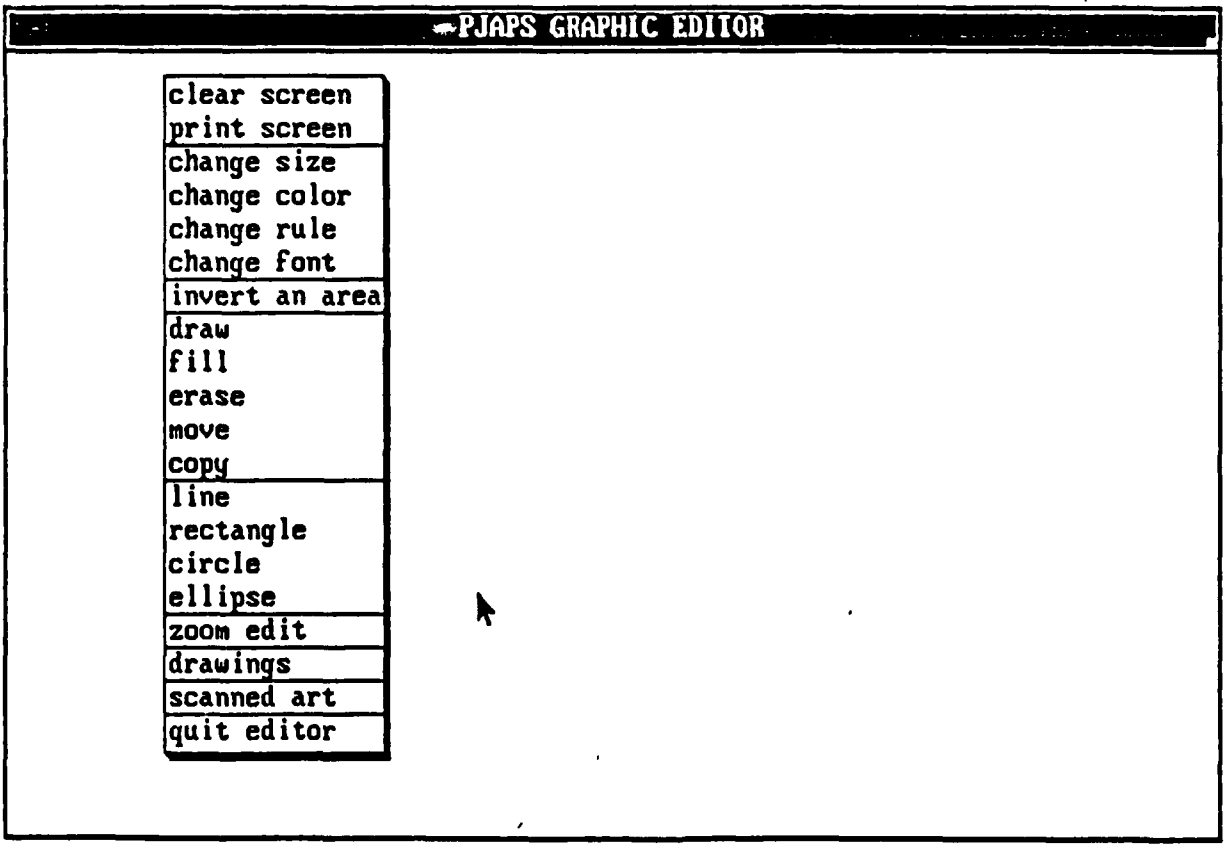
Overview of the PJAPS Graphics Editor

The PJAPS Graphics Editor provides tools for creating original drawings and/or modifying graphics stored in the database to fit a new requirement. Virtually the same tools are used to prepare static, animated and locator graphics. Therefore, this guidebook describes how the graphics editing tools are used to prepare static graphics.

- Select the menu option "create graphics". Another menu will then appear on your screen asking if you want to create static or animated graphics. Select "static" and press the left mouse button.

- Press the right mouse button to call up the menu shown in Table 4. This is the same menu you will encounter when preparing animated graphics, except that the animation menu contains the additional options "copy Position 2" and "animations".

TABLE 4. PJAPS GRAPHICS EDITOR MENU



Graphics Tools

"Drawing" Option. This option supports free hand drawing of figures or portions of figures. This option can be used in association with other tools to create a graphic.

- Select "drawing" option and press the left mouse button.
- To draw, keep the left mouse button depressed as you move the icon around the screen.

Line, Rectangle, Circle and Ellipse Options. The PJAPS graphics editor contains four commonly used geometric shapes for use in preparing your drawings. All of these shapes are drawn by depressing the left mouse button and keeping it depressed as you move the icon around the screen. A large difference among these options is in terms of the effects of movement of the icon.

- The line option draws straight lines only. Movement of the icon gives you control over the length and direction of a line. To begin drawing a line, move the icon to the point where you want the line to start and press the left mouse button. Keep this button pressed until you have completed your line.
- After you select "rectangle" from the menu, the icon will take the form of a right angle. To draw a rectangle, depress the left mouse button and keep it depressed as you move the icon across the screen (i.e., moving the icon changes the shape and size of the rectangle). When you release the left mouse button, the shape of the rectangle will be fixed.
- When you select "circle" from the menu, the icon will take the form of a cross. The center of the circle you draw will be at the location of the cross when you initially depress the left button. Keep the left button continually depressed as you move the icon away from its start point. The further you move the icon before releasing the left mouse button, the larger the resulting circle will be.
- Movement of the icon when using the ellipse option changes the shape and size of the ellipse.

Using Drawing and Clipart Libraries. The task of preparing graphics within JPAPS is facilitated by multiple graphics databases. To gain access to these libraries, select the "drawings" option. A new menu will appear which gives you the options "Save Drawing", "Get Drawing", "Delete Drawing", "Save Clipart", "Get Clipart" and "Delete Clipart".

- The CLIPART portion of the database contains drawings of objects likely to employed within many different graphics. Examples of such objects include screwdrivers, lights, gages/meters. To gain access to clipart, select "get clipart". A clipart menu will then appear on your screen. (NOTE: The clipart menu, like the procedures and graphics menus, contain lists of file names. These menus need to be replaced with ones having names which can be readily understood by the user, such as "screwdriver".) The objects addressed by the current clipart database are somewhat limited and need to be enhanced in the future. You also have the option of creating and storing your own clipart using the "save clipart" option.

- The drawing portion of the database contains figures which are generally more elaborate than "clipart" figures. Within the PJAPS demonstration software, the drawing portion of the database contains the graphics used for the various procedures.

- Graphics can be prepared by combining "drawings" and "clipart". For example, you might prepare a graphic showing a screwdriver in use on a particular item of equipment.

Using the "Zoom Edit," "Erase," "Move," "Copy," and "Invert Area" Tools. These options give you the ability to easily edit geometric shapes, library drawings/clipart, original drawings and scanned art. Directions for using each option are provided below.

- When you select the "zoom edit" option a box will appear on your screen which you can move around and enlarge using the mouse to define the specific portion of your figure to be edited. When the box surrounds the area of interest, press the left mouse button. Figure 6 provides an example of the zoom edit screen with an enlarged portion of a graphic ready for editing. Editing is accomplished by moving the icon to a grid and pressing the left mouse button to add filler to an empty grid or delete filler from a filled grid. When you have finished making edits, move the icon to the box labeled "save", and press the left mouse button.

- Call up the graphics editing menu, move the icon to "erase' and press the left mouse button. The icon will take the form of a tiny eraser. To use the icon as an eraser, keep the left mouse button continually depressed as you move across any art to be erased.

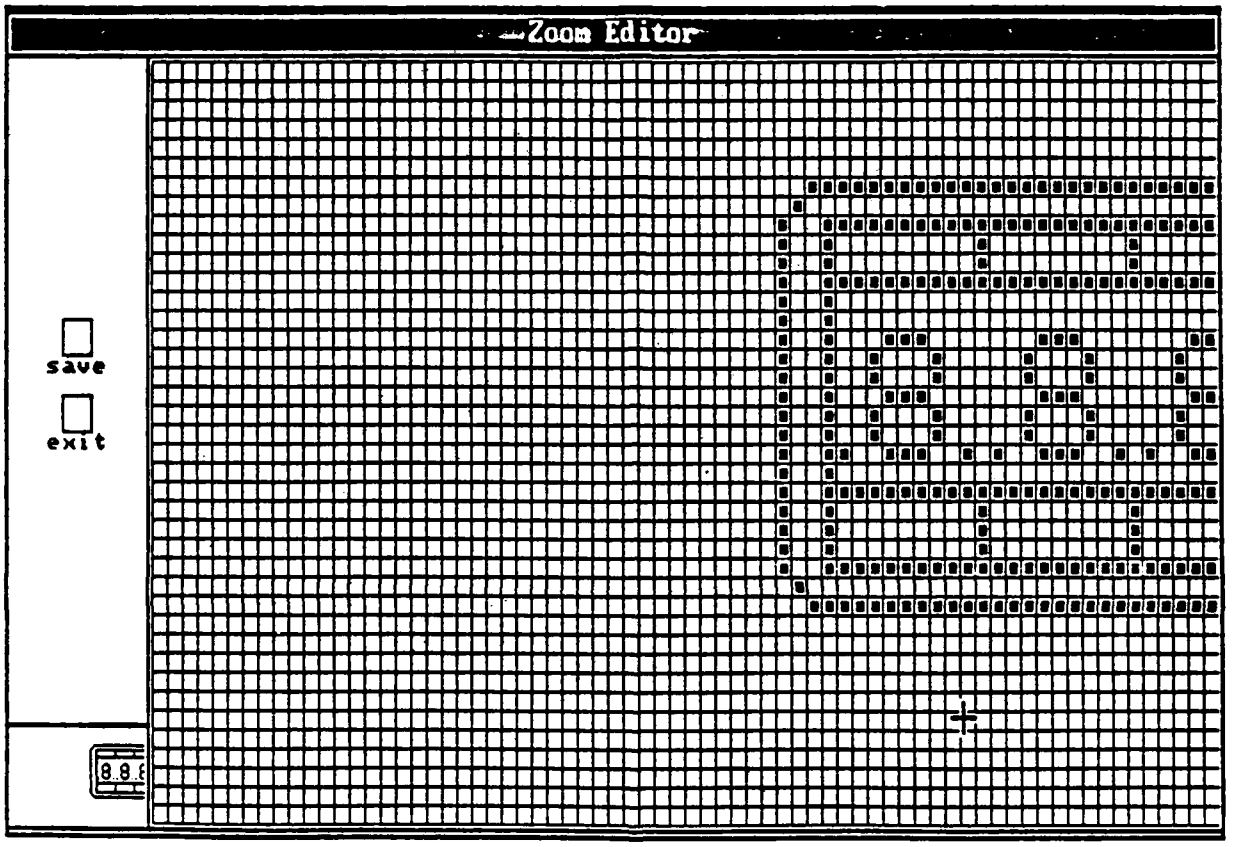


Figure 6. Example of a Zoom Edit Screen

- Call up the graphics menu and select "move". A box with a right angle in the upper left corner will appear on your screen. Move the icon until the angle marks the upper left limit of the figure (or portion of a figure) that you want to move, then press the left mouse button. The angle will then invert and move to the bottom right corner of the box. Move the icon until the angle marks the bottom right corner of the figure to be moved, then press the left mouse button. The icon will then change from an angle to a cross. Move the icon /box until it is in the location where you want to move the figure, then press the left mouse button.

- Call up the graphics menu and select the "copy" option. Procedures for copying a figure or portion of a figure are the same as those for moving a figure, described directly above.

- Call up the graphics menu and select "invert area". (NOTE: Although the process of identifying an area to be inverted should intuitively be the same as that for identifying an area to be copied or moved, a distinctly different process is used to mark an area to be inverted.) The icon will take the form of a right angle (without a box). Move the icon to a point where it marks the upper left boundary of the figure or portion of a figure to be inverted, and begin continuously pressing on the left mouse button as you move the icon down to a point where it marks the bottom right boundary of the area to be inverted. Release the left mouse button when the icon has reached the bottom right corner of the area to be inverted. The area you have identified will then be displayed in reverse video.

Making Hard Copies of Graphics

The "print screen" option in the graphics editor menu is used to make hard copies of drawings produced within PJAPS. It is important to note that these hard copies, like the hard copies of procedures prepared with the procedure editor menu, are produced without the ten minute delay required to make copies of other PJAPS screens.

Savings Graphics/ Preparation of Locator Graphics

Saving Drawings. When you have completed a drawing, call up the graphics editor menu and select "drawings". A second menu will then appear on your screen. Select the option "save drawing". A third menu will then appear listing the names of all drawings in the database and the option "new name". If you select the "new name" option, PJAPS will then ask you to type the title of the new drawing.

Preparation of Locator Graphics. After you have named a drawing in the process of saving it, PJAPS will prompt you to "select the parent of this picture". Press the right mouse button for a menu of all of the drawings on file. Select the file name of a drawing of the larger system in which in which the drawing you have just created is embedded (e.g., if the drawing you have just created is of a particular item of equipment within a control console, then the parent drawing might be one of the entire console). PJAPS will then display the parent drawing and ask you to specify the location of the parentage. Move the icon to the approximate area of the parent drawing where the drawing you have just completed is located, and press the left mouse button.

Requirements Unique to Animated Graphics.

Animated graphics are created using the same tools as static graphics. The process of preparing animated graphics differs from that used in preparing static graphics in the ways listed below.

- Select the "create/modify animated graphics" menu instead of the static graphics menu. The screen will be split into position 1 and position 2.

- Prepare a drawing in position 1, then move the icon to position 2. Call up the graphics editing menu and select the option "copy position 1". A duplicate of the figure in position 1 will appear in position 2.

- Use the graphics editing tools to alter the figure in position 2 to reflect the effects of the required action on the figure in position 1 (e.g., if the required action on the item of equipment in position 1 will result in a changed meter reading, edit the figure in position 2 to illustrate the change).

- Call up the graphics editor menu and select the "animation" option. When the animation menu appears, select the option "save animation". An icon with a box will appear on screen that you can use to mark the location of the figure in position 1. When the frame of the box surrounds the figure, press the left mouse button. (Note: PJAPS will automatically save the figures in positions 1 and 2 by marking position 1).

- PJAPS will then prompt you to provide the information needed to prepare a locator graphic, as described above (i.e., the same procedure is used for both static and animated graphics).

VI. Summary of PJAPS' Strengths and Weaknesses

From the Perspective of Potential Technical Manual Users

PJAPS provides a mechanism whereby the level of detail of information about procedures can be precisely tailored to meet the specific information needs of technical manual users. Users requiring little guidance can simply review tasks at a broad level without getting into the details of performing subtasks. Users requiring more guidance can take advantage of:

- detailed step-by-step procedures for performing subtasks;
- static graphics to help the user identify the parts of an item of equipment that are being described;
- animated graphics to help the user understand the required action on an item of equipment;
- locator graphics to help the user locate an item of equipment embedded in a larger system.

If it were not for one immediate problem, the PJAPS would meet five of the six criteria listed in Table 1 (page 3). The problem is a user unfriendly menu of procedures. The present menu uses a file name for each procedure, and these names are limited by the MS DOS operating system to eight characters. A user-friendly menu providing the full names of procedures, and "pointing" to the file names in a manner transparent to the user, would correct this problem.

A second possible problem is of unknown dimensions. The user of the current PJAPS demonstration software experiences lengthy delays in gaining access to certain procedures. The current delays are annoying, but, on the other hand, thumbing through one or more written technical manuals to gain access to information is also a lengthy/annoying process. The amount of delay/annoyance associated with the demonstration software might represent an improvement over paper-based manuals. However, the amount of information addressed by a complete electronic manual might reasonably be much greater than that addressed by the demonstration software, and one minute delays in gaining access to information might become five or ten minute delays.

The sixth criterion for an effective electronic manual is that of providing an efficient browsing capability. The browsing capability of the PJAPS is limited. Frames marked or stored are not retained in memory when the user exits a particular procedure. However, this problem is attenuated by the linkage feature of the PJAPS. The linkage feature brings detailed information to the user within a procedure without the user having to leave the procedure and go elsewhere for this

information. Compare this with a situation where the user of a paper-based manual begins to browse through a section of the manual and is forced to leave that section to go to other sections (or other manuals) to seek out references. Whether or not the existing browsing capability of PJAPS is adequate is an issue which needs to be addressed by future research.

From the Perspective of Potential Authors of Electronic Manuals

The PJAPS demonstration software provides clear/concise information about what types of information must be developed to prepare procedures in the electronic format. Producing an electronic manual in the PJAPS format requires authors to

- identify and describe procedural steps,
- identify cases where one procedure is a subtask within another more complex procedure,

and

- identify or develop static and animated graphics which help to clarify specific procedural steps

The job of using this input to develop the array of screens required for effective display of technical information is appropriately left to PJAPS algorithms.

Overall, the PJAPS authoring system makes it easy to input the required information. However, there are areas which require improvement.

The major problem confronting authors is the use of procedure and graphic file names for menus. As the numbers of procedures and graphics increases, the ability of authors to remember these file names is severely strained. Menus with complete procedure/graphics titles would make the PJAPS much easier to use.

The PJAPS graphics production process is friendly to users, with the exception of a lack of standardization in the use of a mouse to prepare graphics. Virtually identical tasks (e.g., identifying a figure or portion of a figure on which an action is to be taken) are performed using the mouse in different ways (e.g., identifying the top left and bottom right boundaries of an area with two quick presses of a mouse button versus holding the mouse button down continually to mark the boundaries of an area). Brief screen prompts explaining how the mouse is to be used for various applications might address this problem.

APPENDIX E
USER'S GUIDE FOR PROCEDURE SIMULATION PRODUCED BY REHEARSAL (PROSPR) DEMONSTRATION SOFTWARE

CONTENTS

CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF TABLES

USER'S GUIDE FOR PROCEDURE SIMULATION PRODUCED BY REHEARSAL (PROSPR) DEMONSTRATION SOFTWARE

I. Introduction to the Procedure Simulation Produced by Rehearsal (PROSPR)

Intended Benefits of PROSPR

PROSPR demonstration software was produced as part of a larger effort to develop formats for electronic delivery of technical information. Such information would be delivered on an Electronic Information Delivery System (EIDS) or a Militarized EIDS (MEIDS). PROSPR is an interactive/electronic tool designed to provide personnel with practice in performing operation and maintenance procedures. In addition, PROSPR is an authoring system used to develop practice sessions. An overview of the PROSPR training delivery and authoring systems is provided in Figure 1.

PROSPR is intended to employ low-cost equipment simulations to provide personnel with training in such procedural tasks as locating equipment components, quickly assessing equipment status and operating equipment. The maximum benefits of PROSPR are gained when the tasks to be trained warrant practice to the point of overlearning (e.g., emergency troubleshooting procedures) and can be trained on operational equipment only at great expense.

The PROSPR training delivery (and authoring) system has the flexibility to adapt training to the skill level of the user. For a given procedure, the user has the option of initial or advanced practice sessions. Initial sessions provide more cues and prompts to guide personnel than do advanced sessions. Further, the initial practice sessions can be easily repeated to gain additional practice. The PROSPR system can also be used to provide repeated practice on a task over diverse situations. That is, the specifics of the situations to be assessed and the actions to be taken to correct a situation can be automatically varied within a given procedure.

The training system necessary to effectively address the types of training needs mentioned above must meet certain requirements. First, the system must provide quality drawings of complex items of equipment and their component parts. Second, drawings, must include "smart objects" such as switches,lights and valves which "work". Third, the system must provide feedback to personnel. Fourth, the cost of producing quality drawings, integrating "smart objects" and implementing a feedback mechanism must be kept low to ensure that the overall system is affordable.

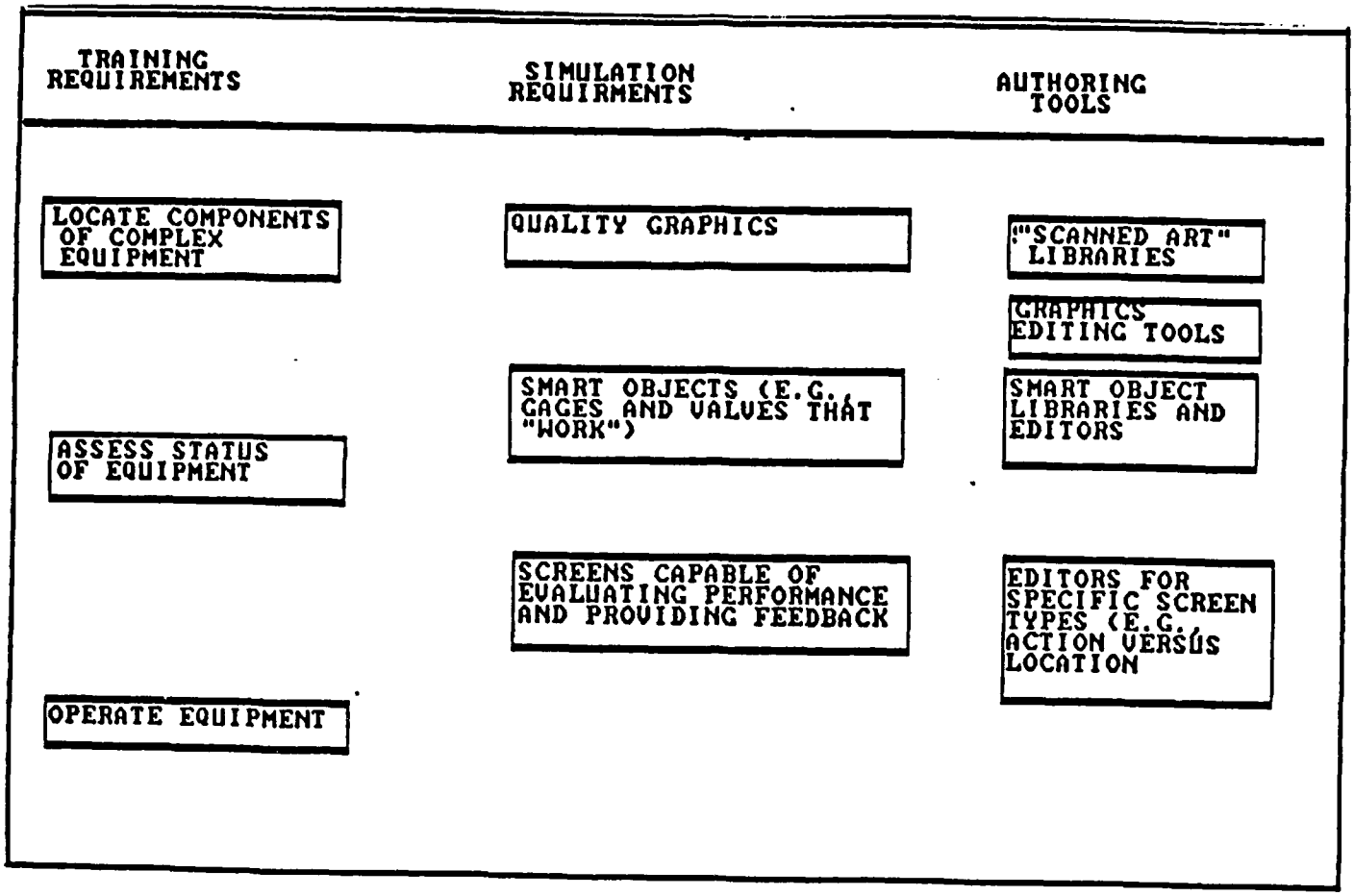


Figure 1. Overview of PROSPR Simulation Delivery and Authoring System.

PROSPR authoring software employs three major design features intended to keep the cost of producing procedure simulations at an affordable able level. First, PROSPR allows authors to use drawings and pictures which have been scanned into a database with an optical character reader. These drawings and pictures can then be refined using graphics editing tools included in the PROSPR software. Second, PROSPR contains a smart object library and smart object editors designed to help authors prepare animated portions of procedural screens (e.g., gages, lights). PROSPR graphics editing tools can be used to modify smart objects contained in the library to fit a particular requirement. Third, it includes editors tailored to the production of specific types of screens (e.g., a location screen editor versus an action screen editor).

Purpose of PROSPR Demonstration Software

PROSPR demonstration software contains specific examples of the application of PROSPR to training the task of "perform computer self test". This sample will give you an opportunity to decide whether PROSPR screen formats and procedural simulations meet the information needs of a diverse user group. In addition, this software also contains a prototype authoring system which might be used to create procedures. A first hand review of PROSPR authoring tools will help you to decide whether these tools are "user friendly" and employ cost-effective procedures to create the diversity of frames required to effectively employ the concept of procedure simulation.

It is important to emphasize that PROSPR demonstration software is expected to be refined during an upcoming trial application. The delivery and authoring software you will use in the demonstration package were developed as a proof of concept test based upon limited applications.

II. Installation of PROSPR Software

Hardware/Software Requirements

- PROSPR software will run on an IBM or IBM compatible PC/XT or AT with a 20 megabyte hard disk drive. However, you should expect long delays when using any system below the AT level.

- The system using PROSPR software should have a VGA graphics adapter for best results.

- The use of PROSPR requires a Microsoft-compatible mouse.

Installing Software on a Hard Disk

PROSPR software is provided on nine floppy disks. The recommended procedures for loading PROSPR software are as follows.

Disks 1-7

- Create a directory called "ST5" (Type MD ST5), and change to the ST5 directory (Type CD ST5).

- To load disks 1 through 7, type RESTORE A: C:\ST5*.* /F (NOTE: Be sure to leave a space between the last asterisk and the forward slash F). The system will then prompt you to load each disk in turn.

Disk 8

- Create a directory called "smallobj" (Type MD SMALLOBJ).
- Change to the smallobj directory (Type CD SMALLOBJ)
- Create a subdirectory called "procs" (Type MD PROCS), and change to the PROCS subdirectory.
- Type COPY A:*.* C:
- Load disk 8 in response to screen prompt.

Disk 9

- Change to the "smallobj" directory
- Create a subdirectory called "scanner" and change to the scanner subdirectory.
- Type COPY A:\SCANNER*.* C:
- Change to the "smallobj" directory again, and create a subdirectory called "backdrop".
- Type COPY A:\BACKDROP*.* C:
- Change to the "smallobj" directory a third time, and create a subdirectory called "smartobj".
- Type COPY A:\smartobj*.* C:

III. Running the Sample PROSPR Procedure (M-1 Tank Computer Self Test)

Starting the Procedure

- Change to the "ST5" directory and type V.

- When the title screen appears, press any mouse button for a menu. Select the option "run a procedure". You will experience a delay of approximately five minutes while the procedure is loaded into memory (NOTE: The amount of time required to load a procedure is a problem which needs to be addressed during refinement of the PROSPR. Keep in mind that the demonstration software is based upon a single procedure, while a mature PROSPR might contain hundreds of procedures.

- The next frame which appears on your screen will contain the title of the procedure, and you will be given the option to run the procedure or quit. The subsequent frame is entitled "System Configuration" and it describes the status of relevant equipment when you initiate the procedure of "performing a computer self test" (i.e., gunner's station is powered up and main gun is forward). This information tells you the context in which you will apply what your are learning during the procedure.

- Press any mouse button to move on to the "mode of practice screen". This screen will ask you to choose between "initial practice" and "advanced practice".

Initial Practice

- Select the initial practice option from the mode of practice screen menu. You will experience a delay of roughly two minutes before the next screen appears.

- The next screen lists the first step in the procedure at the top (i.e., "power up gunner's station". In this case, the portion of the procedure addressed by the sample procedure begins after this step, and the message "gunner's station is already powered up" appears at the bottom of the screen. Press any mouse button to move to the next screen.

- The top of the next screen states the procedural step which follows: "Make sure hydraulic gage shows 1500 to 1700 psi. If not, notify organizational maintenance." (see Figure 2.) This screen falls in the generic class of "location" screens. A message on one side of the screen asks you to locate the hydraulic gage, and the other side of the screen provides a picture of the gunner's station. You as a user

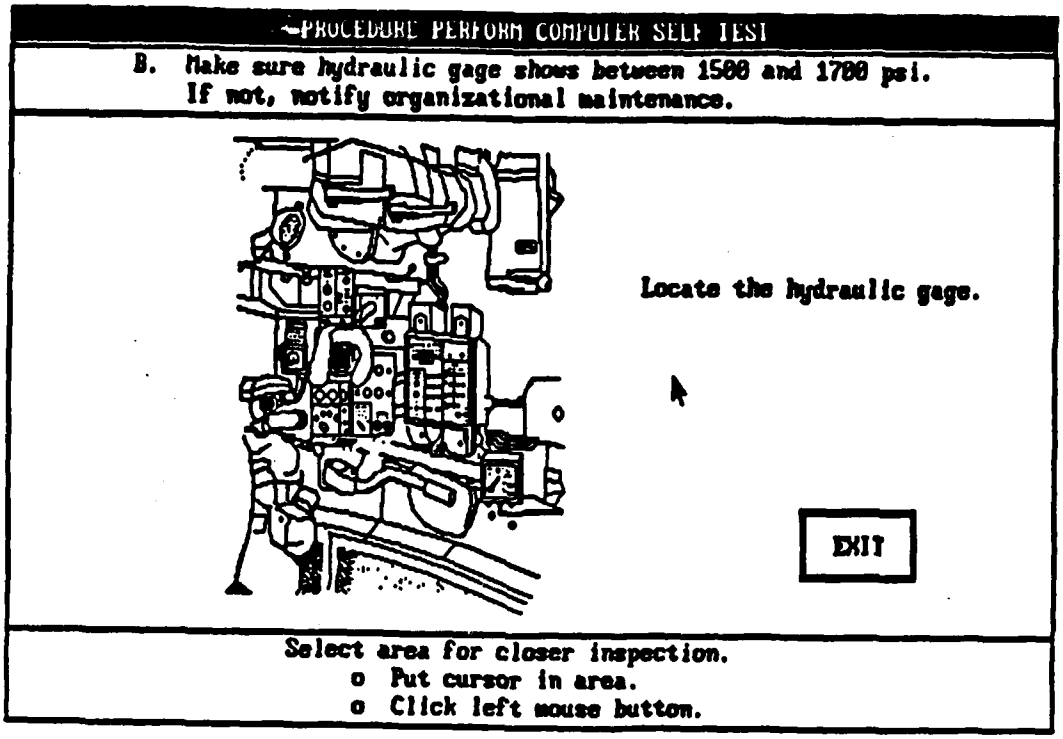


Figure 2. Example of a PROSPR "Location" Screen Used to Train Personnel to Locate Components of Complex Equipment

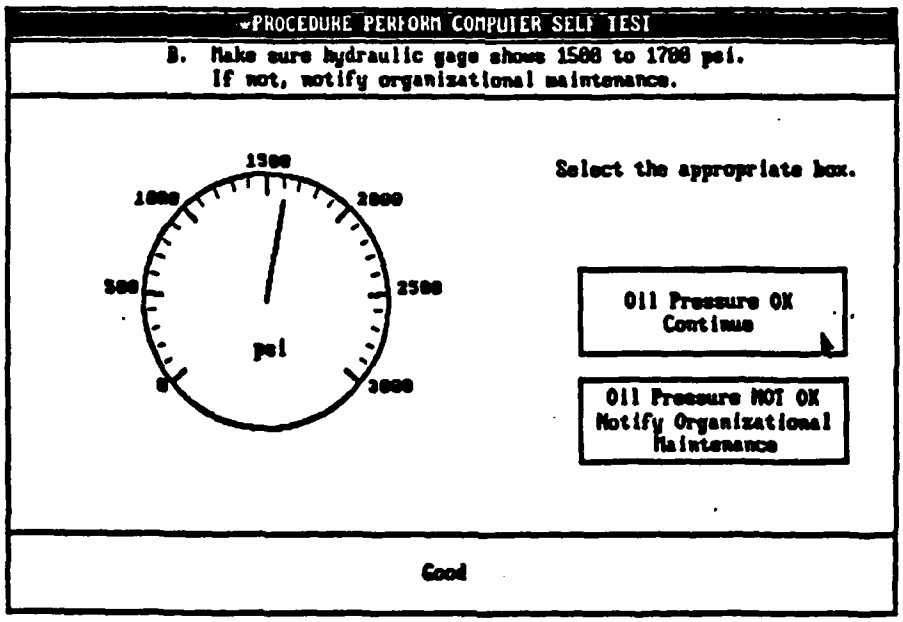


Figure 3. Example of a PROSPR "Action" Screen Requiring Personnel to Make a Judgement about the Status of Equipment.

are expected to move the icon to the area where the hydraulic gage is located and press the left mouse button.

- If you have selected the correct location, the message "good will appear at the bottom of the screen

- If you have selected the wrong location, the correct location will quickly "flash", and the message "try again" will appear at the bottom of the screen.

- After you have correctly located the item of equipment, press any mouse button to move to the next screen. The next screen will repeat the statement of the procedural step and give you an opportunity to practice the application with a simulation (See Figure 3). This screen falls in the generic class of "action" screens. A gage will appear on your screen along with two response blocks. One block reads "Oil pressure okay..continue" and the other reads "Oil Pressure Not Okay...Notify Organizational Maintenance". Read the gage and move the icon to the appropriate response block and press any mouse button. In this screen, the gage is a "smart object" which has been programmed to vary in terms of readings. If you make the correct choice, a "good" message will appear at the bottom of the screen, otherwise you will receive a "try again" message.

- You cannot go on to the next screen until the pressure reading is in the safe range. Therefore, the screen will prompt you to reset the gage by pressing any mouse button. You must also respond to the correct response block after the gage is reset. If the gage reading is still in the unsafe range, you must repeat the entire procedure. When you have met all of the requirements, press any mouse button to move to the next screen.

- As you move through the next few screens you will again be introduced to new procedures, asked to locate items of equipment and asked to judge whether simulated equipment meets certain criteria. Near the end of the procedure, you will encounter a new type of practice exercise in which you are asked to act upon simulated equipment to correct a situation. Unlike the previous example, this practice involves the use of two "smart objects". In this case, you must adjust a simulated "normal mode drift" knob to stop main gun and turret drift. Figure 4 shows a screen which allows you to practice adjusting a knob, and such screens are referred to as "note" screens. Figure 5 shows the screen you would use to adjust a normal mode drift knob (one smart object) to stop main gun and turret drift as simulated in the site picture of the gunner's scope (the second smart object). This particular screen

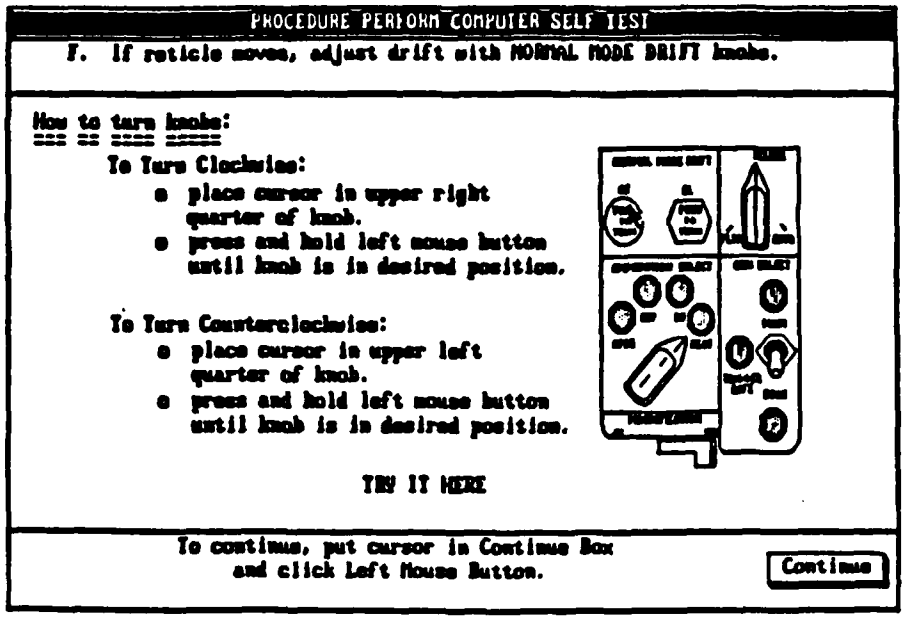


Figure 4. Example of a PROSPR "note" screen which gives personnel the opportunity to practice manipulating simulated equipment.

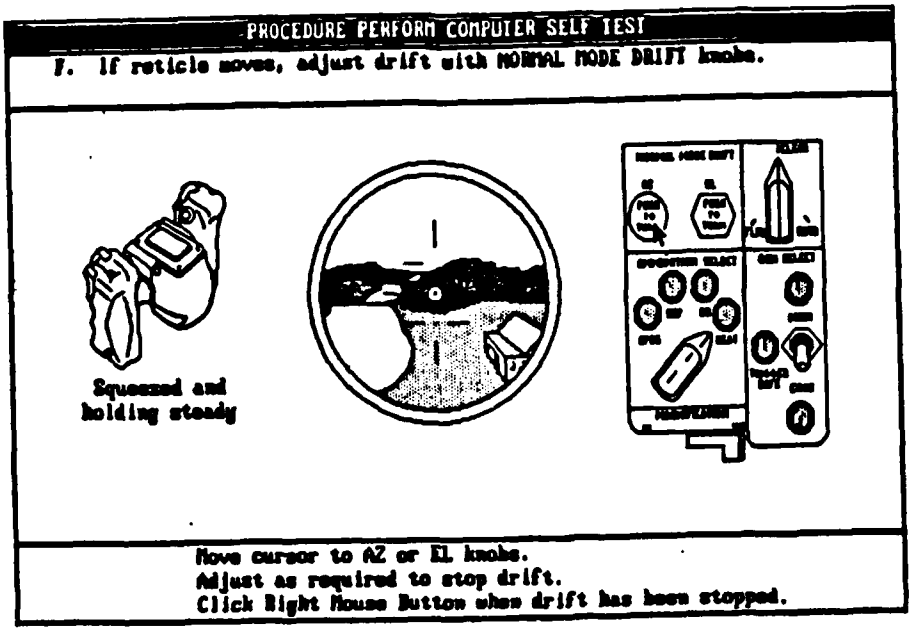


Figure 5. Example of a PROSPR "action" screen requiring personnel to use simulated equipment to correct a maintenence/operation problem in related equipment.

cannot be fully appreciated by viewing a static picture, because the site picture in the PROSPR software is animated.

- When you reach the end of the procedure, you will be given the option of continuing on to the next procedure or repeating practice on the current procedure. Only one procedure has been addressed in the PROSPR demonstration. If you select the option to move to the next procedure, you will receive a message that you have reached the end of the procedures. Press the right mouse button to call up a menu that will allow you to exit PROSPR.

Advanced Practice

Advanced practice differs from initial practice in terms of the features listed below.

- Advanced practice does not provide procedural steps or other guidance.

- Advanced practice does not cue the user to the correct location of an item of equipment until the user has failed three times to identify the location.

- The advanced practice exercise does not include a menu which allows the user to repeat an exercise.

- Advanced practice does not include the use of "note" screens.

IV. Overview of the PROSPR Authoring Process

Developing a Program of Instruction (POI) Which can be Applied with PROSPR.

Preparing a POI which can be applied through PROSPR is a surprisingly simple, clear cut job. Preparing a POI requires you to:

- Specify the items of equipment you want to train personnel to locate quickly,
- Specify the larger equipment system within which these items are to be located,
- Decide what you want personnel to be able to check/ assess for each item of equipment,
- Decide what items of equipment you want personnel to train to operate, and specify what this operation involves (e.g., adjusting a knob to calibrate a gage).

Three Phases of Authoring with PROSPR

The job of creating PROSPR procedures is divided into three parts as described below.

- Select or Develop Smart Objects Used within the Various Screens. Smart objects are those which directly or indirectly respond to the actions of the user (e.g., switches, lights). PROSPR includes libraries of frequently used objects which you can tailor to fit a particular requirement. PROSPR also provides a variety of graphics editing tools to help you accomplish these modifications. (NOTE: The current library of smart objects is still limited and will be enhanced as the software is refined).
- Develop the Backdrops for Screens. Backdrops can include all of the figures and text used in a screen except the smart objects. You can use PROSPR to create either monochrome or color backdrops. In preparing these backdrops, you can also make use of scanned art (i.e., pictures which have been stored in the database through the use of an Optical Character Reader). When creating backdrops, you will be assisted by many of the same graphics aids used in editing smart objects.
- "Design" and Sequence the Various Screens. In this phase of the authoring process, you will design the variety of screens which can be included in a PROSPR procedure, and each type of screen has its own editor menu to assist you.

Special editing menus are available for locator screens (screens asking the user to locate a particular item of equipment), action screens (screens requiring the user to assess a variable situation or perform an action), text screens, menu screens and screens with figures (other than locator/action screens). The work of preparing these screens is reduced by the fact that many of the same smart objects and backdrops are used in more than one screen.

Scanned Art and Smart Object Libraries

PROSPR supports the use of pictures/figures which have been scanned into the database (through the use of an optical character reader) and saved in Microsoft Paint format. These pictures form the raw materials from which most of the PROSPR simulations can be created. Modifications of these pictures to meet the needs for a particular screen are facilitated by the use of graphics editing aids embedded in the PROSPR software. Once these pictures have been modified, the modified versions are stored and become available for future use as part of an evergrowing library. It is important to note that the size and content of the scanned art library and libraries of modified pictures are under the complete control of the authors using PROSPR.

PROSPR also provides "smart object" editors and libraries of frequently required "smart objects". Smart objects are those specific portions of simulations which need to be assessed by students or manipulated by students. In screens were the student must act upon one portion of a simulation to influence another portion, two smart objects are involved. Each class (e.g., switches) and certain subclasses (e.g., 3-position switch) of smart object editors have their own unique editing systems, and these editors are created by the developer of PROSPR authoring software. This authoring software also contains libraries of pictures of smart objects falling within various classes and subclasses. Authors using PROSPR can modify the pictures of smart object contained in the library using the graphics editing tools described below and save these modified smart objects within an ever growing smart object library.

Graphics Editing Aids Used in Preparing Smart Objects and Backdrops.

In creating smart objects and backdrops, you will use a menu of core graphics tools. In addition, there are specific function tools tailored to meet specific objectives. For example, there is a three-position switch editor and a separate two-position switch editor to help you create switches for use as smart objects. The tools contained in the core editing menu are listed in Table 1 and described below.

TABLE 1. CORE EDITOR MENU

Clear Screen
Change Size
Change Rule
Change Font
Draw
Fill
Erase
Move
Copy
Line
Rectangle
Circle
Ellipse
Zoom Edit
Pictures
Backdrops
Scanned Art
Exit Editor

"Clear Screen" Option. Selection of this option clears the screen of all text and graphics. This option allows you to begin work on a new backdrop after you have completed and saved another backdrop without exiting the backdrop editor. In addition, this option proves useful in cases where the backdrop you are preparing is unsatisfactory and you want to start over.

"Drawing" Option. This option supports free hand drawing of figures or portions of figures. This option can be used in association with other tools to create a graphic. Select "drawing" option and press the left mouse button. To draw, keep the left mouse button depressed as you move the icon around the screen.

"Line," "Rectangle," "Circle," and "Ellipse" Options. The PROSPR graphics editor contains four commonly used geometric shapes for use in preparing your drawings. All of these shapes are drawn by depressing the left mouse button and keeping it depressed as you move the icon around the screen. A large difference among these options is in terms of the effects of movement of the icon.

- The line option draws straight lines only. Movement of the icon gives you control over the length and direction of a line. To begin drawing a line, move the icon to the point where you want the line to start and press the left mouse button. Keep this button pressed until you have completed your line.

- After you select "rectangle" from the menu, the icon will take the form of a right angle. To draw a rectangle, depress the left mouse button and keep it depressed as you move the icon across the screen (i.e., moving the icon changes the shape and size of the rectangle). When you release the left mouse button, the shape of the rectangle will be fixed.

- When you select "circle" from the menu, the icon will take the form of a cross. The center of the circle you draw will be at the location of the cross when you initially depress the left button. Keep the left button continually depressed as you move the icon away from its start point. The further you move the icon before releasing the left mouse button, the larger the resulting circle will be.

- Movement of the icon when using the ellipse option changes the shape and size of the ellipse.

"Zoom Edit," "Erase," "Move," "Copy," and "Fill" Options. These options give you the ability to edit geometric shapes, library drawings/clipart, original drawings and scanned art easily. Directions for using each option are provided below.

- When you select the "zoom edit" option a box will appear on your screen which you can move around and enlarge using the mouse to define the specific portion of your figure to be edited. When the box surrounds the area of interest, press the left mouse button. Figure 6 provides an example of the zoom edit screen with an enlarged portion of a graphic ready for editing. Editing is accomplished by moving the icon to a grid and pressing the left mouse button to add filler to an empty grid or delete filler from a filled grid. When you have finished making edits, move the icon to the box labeled "save", and press the left mouse button.

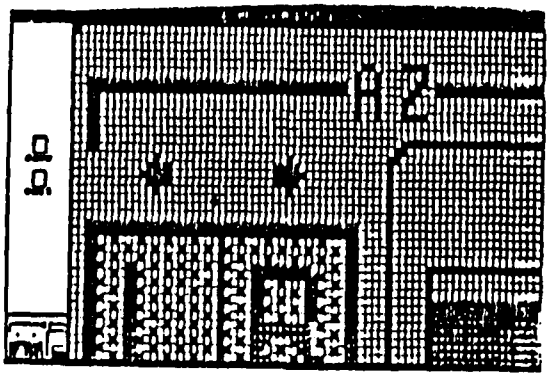


Figure 6. Use of the "Zoom Edit" Feature to Modify Scanned Art or Original Drawings.

- Call up the graphics editing menu, move the icon to "erase' and press the left mouse button. The icon will take the form of a tiny eraser. To use the icon as an eraser, keep the left mouse button continually depressed as you move across any art to be erased.

- Call up the graphics menu and select "move". A box with a right angle in the upper left corner will appear on your screen. Move the icon until the angle marks the upper left limit of the figure (or portion of a figure) that you want to move, then press the left mouse button. The angle will then invert and move to the bottom right corner of the box. Move the icon until the angle marks the bottom right corner of the figure to be moved, then press the left mouse button. The icon will then change from an angle to a cross. Move the icon /box until it is in the location where you want to move the figure, then press the left mouse button.

- Call up the graphics menu and select the "copy" option. Procedures for copying a figure or portion of a figure are the same as those for moving a figure, described directly above.

- The "fill" option allows you to quickly fill in an area bounded by lines. Select the "fill" option. Move the icon to the area to be filled in, and press the left mouse button.

V. Selecting/Designing and "Auditioning" Smart Objects

Overview of Smart Object Editors in the Auditoner Menu

Select the "auditioner" menu. When the screen title has changed to auditioner, press the right mouse button for a menu. The menu will provide a list of classes of smart objects (sèe Table 3). For each type of smart object there is at least one editing system that will help you to work on a specific object type. For example, there are switch editors versus gage editors.

TABLE 2. SMART OBJECT EDITORS CONTAINED IN THE PROSPR DEMONSTRATION SOFTWARE

Switches
Lights
Valves
Gages
Digital Display
Knobs
Scopes

You have the option of editing one of the smart objects contained in the database or creating a smart object that falls within one of these classes with the aid of class-specific editing tools. As you will find in reviewing the information provided below, the protocols for using the editors differ markedly among classes of objects. (NOTE: Differences among the various smart object editors in terms of menus and procedures is a problem which needs to br addressed in future refinement of PROSPR).

Your first menu options after selecting one of the classes of smart objects gives you the options of, for example, "getting a switch" or "designing a switch". When you "get" a smart object from the database, you can "audition" the object, but you cannot modify it. "Auditioning" the object means testing it to make sure that it meets your requirements for a smart object. When you select the "design" option, you can gain access to a library of smart objects and use graphics editing aids to modify the appearance of these objects.

"Switch" Editor

- Select the switch option from the auditioner menu and press the left mouse button. Select the option "design a switch" from the next menu. The third menu will give you the option of designing a 2-, 3-, 4- or 5-position switch. Select the 3-position switch option.

- Your screen will then be divided in three panels labeled "Position 1", "Position 2" and "Position 3". Note that the top of one panel is labeled in blue on a yellow background, while the other panels are labeled in yellow on a blue background. You may only work on one panel at a time, the panel with the yellow background. Also note that you can change the active panel by moving the icon to the top of an inactive panel and pressing the left mouse button.

- Press the right mouse button for a switch editing menu. This menu will provide the same graphics editing tools as the core menu (e.g., "erase", "zoom edit"). In addition, this menu contains the option "switches" which is used to call up a menu with the options of "getting", "saving" or "deleting" a switch. In cases where a switch are in the database might meet your requirements with minor graphics editing, you can select the desired switch from the "get switch" menu.

- To see whether the smart object you have designed functions in the intended manner, you will need to exit the "design a switch" menu and select the "get a switch" menu. A menu of switch options will appear. Select the "3-position switch" option. The next menu contains the names of the 3-position switches in the database. Select the switch you wish to audition, and press the right mouse button for an auditioning menu. This last menu will allow you to view the smart object in various postions.

Light, Valve, Digital Display and Knob Editors

These four editors work in the same way as the switch editor described above. It is important to point out that the PROSPR demonstration software does not contain any valves. However, you can use PROSPR software to create your own valves (see Figure 7).

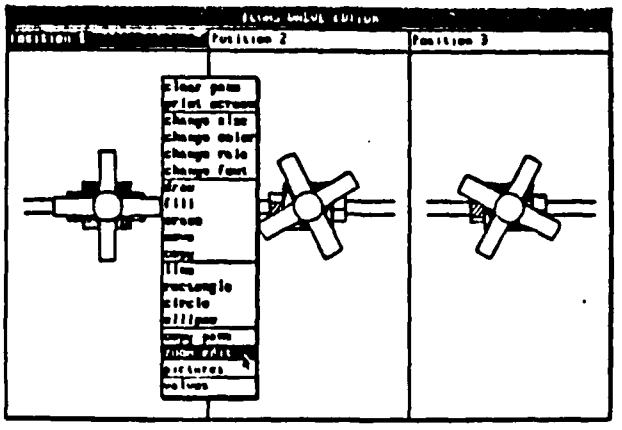


Figure 7. Example of a PROSPR smart object editor.

Gage Editor

The gage editor differs considerably from the smart object editors described above. When you select the option "design a gage", a menu will appear with the options "clear screen", "generate gage", "generate meter", "files" and "exit editor". The "files" option allows you to delete a gage or save a gage, but it does not allow you to "get" a gage. To create a gage you must select the option "generate gage" or "generate meter".

When you select the option "generate gage", the icon will take the form of a right angle. Press the left mouse button and keep it depressed as you move the icon on screen. The figure created will be a circle.

PROSPR will then ask you to label the gage begining with the lower left hand. Type each label and then skip a space before typing the next label. For example, to create a gasoline gage, you might type "E 1/4 1/2 3/4 F". PROSPR will then prompt you to indicate the low value of the gage and then the high value of the gage (both in integers).

Next, PROSPR will ask how many regions are to be placed between tick marks (i.e., "tick marks refer to the low and high end of the gage..E and F in the example of the gasoline gage above). PROSPR requires that there be at least two regions between each label. For example, if there are five labels, then there must be at least eight regions between "tick marks".

Select the "file" menu to save the gage you have created. Give the gage a name with eight characters or less. Exit the gage editor, press the right mouse button to call up a menu, and select the option "get gage". A menu of the gages in the database will then appear. Select the gage to be auditioned.

After you have called up the gage to want to check, press the right mouse button for the gage auditioner menu. This menu will contain the options listed and described below.

- Place Needle. This option allows you to set the reading on the gage. Move the icon to the point on the gage where you want to place the needle, and press the left mouse button. The needle will move to the selected location.

- Auto Increase. This option will cause the gage reading to increase at a rate you select. PROSPR will ask you to indicate the number of seconds between increments.

- Auto Decrease. This option will cause the gage reading to decrease at a rate you select.

- Set Movement Rate. This option allows you to specify the number of seconds between increments (or decrements)

Scope Editor

- When you select the "scope" and then "design a scope" options, your screen will be divided into an upper portion labeled "External View of Scope" and a lower portion labeled "Scope Contents". The upper portion is where you will design the lens of the scope, and the lower portion is where you will design the view (e.g., terrain) which the student observes when looking through the scope.

- Only one portion of the screen is active at a time (the portion with the blue frame on top). To change activity from one portion to another move the icon to the top of the active portion and press the left mouse button. Call up the scope design menu by pressing the right mouse button. Select the "scopes" option to call up a menu of "get a scope", "save a scope" and "delete" a scope options. Select the "get a scope" option, to call up a menu of scopes in the database. "Erasame" is the only scope currently in the database, and the drawing called up will depend upon whether the upper or lower portion of the screen is active. If the upper portion is active, the drawing will be that of the lens of a scope. If the lower portion is active, the drawing will be of a terrain picture. Both drawings can be modified, using graphics editing tools and saved under a new name.

- Exit the scope editor, then call up the scope auditioner menu. The options for this menu are as follows: "scope drift", "move up", "move down", "move left", "move right" and "new object". This option is used to select the site picture the student will see. Select the drift option will cause the view seen by the student to drift. Selecting one of the other movement options will cause the site picture to move until you press the right mouse button to freeze the view on a particular portion of the drawing.

VI. Creating Backdrops

Your Goals in Creating a Backdrop Library.

Many of the backdrops you create will be used in more than one screen, and certain of the screens you create will contain more than one backdrop. The major candidates for backdrops are static figures to be included in your screens. Such static figures would include static portions of animated figures. For example, you might plan to use an animated figure in which a knob on a control console can be adjusted to change a reading on a meter. The knob and the meter are smart objects, and the remainder of the control console is a backdrop.

You can create many of your backdrops from figures which have been scanned into the database through the use of an Optical Character Reader (OCR). PROSPR editing tools can help you edit this "scanned art" to make necessary revisions.

- Select the "Backdrop" option from the title screen menu. A few minutes will be required for this part of the program to be loaded.
- The next screen will give you the option of a "Monochrome Backdrop Editor" or a "Color Backdrop Editor". Select the monochrome editor.

Monochrome Backdrop Editor

The monochrome backdrop editor menu contains five options in addition to the core graphics editing options listed in Table 1. These new options are listed and described below.

- Change Mask Option. This option can be used to change the shade in which text and drawings are created. After you select this option, a menu will appear offering the options of white, light gray, gray, dark gray and black. The mask option, does not affect drawings and texts which have already been created. After selecting the option of, for example, light gray, then the figures/text you create will be light gray.

- "Pictures" Option. This option is used when you want to gain access to a picture library, save a picture you have created in the picture library or delete a picture from the library. After you select the "pictures" option, a menu will appear to ask if you want to save a picture, get a picture or delete a picture. PROSPR demonstration software does not contain any "pictures" in its database. You can create a quick picture and then save it to create your own demonstration. When you save a picture, give it a name with eight letters or less. After you

have saved a picture, you can use the "get picture" option to call up the option you have saved and the "delete picture" option to remove it from the database.

- "Scanned Art" Option. A "scanned art" library is included in the PROSPR demonstration software. After you select this option, a menu will appear with the file names of the scanned art provided in the database. Move the icon to the title of the figure you want to select, and press the left mouse button.

- "Backdrops" Option. The backdrops option is used to store, delete and gain access to backdrops.

Color Backdrop Editor

Options unique to the "Color Backdrop Editor" or used in a different fashion with this editor are as listed and described below.

- Change Color Option. This option works much the same way as the "masking" option for the monochrome backdrop editor. Selection of the color option will provide a color bar menu allowing you select among sixteen colors/hues.

- Change Halftone Option. When you select this option, a menu will appear giving you the options of "dark grey", "light grey" and "normal". Selection of the dark grey option in combination with the red color option will result in a situation where half of the dots composing a line are red and half are dark grey.

- Zoom Edit Option with Color Menu. The zoom editor which can be accessed through the color backdrop editor differs from the standard zoom editor by providing a pallete of sixteen colors on the left margin of the screen. Move your icon to the color you want to use and press the left mouse button.

VII. Using the Rehearser to Create Screens

Overview of the Rehearser

The rehearser portion of PROSPR software provides editors which help you to create specific types of screens. That is, certain tasks you would otherwise have to perform to create a particular type of screen is done for you by the editor. In creating a location screen, for example, your job is to

- call up the backdrop or backdrops you want to use from the backdrop menu,
- use your mouse to place a rectangle around the area to be located by the "student",
- type the instructions to be given to the student (e.g., "locate the gunner's primary site"
  and
- type the procedural step at the top of the screen.

The rehearser "performs" the additional tasks required to provide feedback to the student. If the student correctly locates the area, the message "good" appears at the bottom of the screen. If the student selects the wrong area, the message "try again" appears at the bottom of the screen and the correct area "flashes".

Most types of screens are prepared in three phases. In the Placement Phase, you will be placing the backdrops and smart objects on the screen, placing smart objects on backdrops (e.g., replacing inert knobs with smart knobs), marking objects/ regions to be located by students and typing text. In the Dependency Phase you will be defining relationships between smart objects for screens where a student's manipulation of one object is intended to influence another object. In the Rehearsal Phase, you will define the staring status of smart objects, type in "prompts" (e.g., instructions to the student about what to do in reacting to a screen), and try out the screen from the perspective of a student.

Embedded, Optional Prompts for Use by PROSPR Authors

The "rehearsal" portion of PROSPR, unlike the other portions of PROSPR authoring software, provides embedded "training" for selected authoring tasks. Upon entering the rehearsal portion of the authoring software you will be given the option to select either "I want prompts during rehearsal" or "I do not want prompts during rehearsal". These prompts are useful the first few times you use the reheaser, but their presence becomes annoying and disruptive when they are no longer required. Fortunately, when you reach the point where the prompts are annoying and unnessary, you can exit the rehearser and reenter without selecting prompts.

Gaining Access to the "Define a Screen" Menu

- Select the rehearser option from the initial PROSPR menu.
- The next menu will ask if you want to "create a procedure", "modify a procedure" or exit rehearser. Select "create a procedure". (NOTE: If you select the modify a procedure option, a new "menu" will appear with the name of the only procedure currently contained in the demonstration software, "conduct computer self test". Selecting this option will result in massive disk swapping and delays, and so it is recommended that you do not select the "modify a procedure" option. This problem in using the "modify a procedure" option should be corrected as the software undergoes further development).
- You will be prompted to provide a formal name for the procedure. Type the full name of the procedure.
- The screen will then ask whether you want help prompts during rehearsal. Select the prompt option.
- The next menu will give you the options "define a screen" "another procedure" or "quit rehearser. Select "define a screen". The next menu will give you option to create the following screens:

  -Text
  -Menu
  -Invisible menu
  -Note
  -Location
  -Action

Creating Location Screens

- Figure 8 is the sample location screen to be created.
- Select the "location" screen option from the "define a screen" menu.
- Type in procedural steps in the "step pane" at at the top of the screen. Notice that the screen title indicates that you are in the "placement phase". Also notice that the screen is automatically numbered by PROSPR. Each screen you create while in a given procedure will be automatically labeled according to type (e.g., location screen) and each type will be numbered in sequence (e.g., location 003).

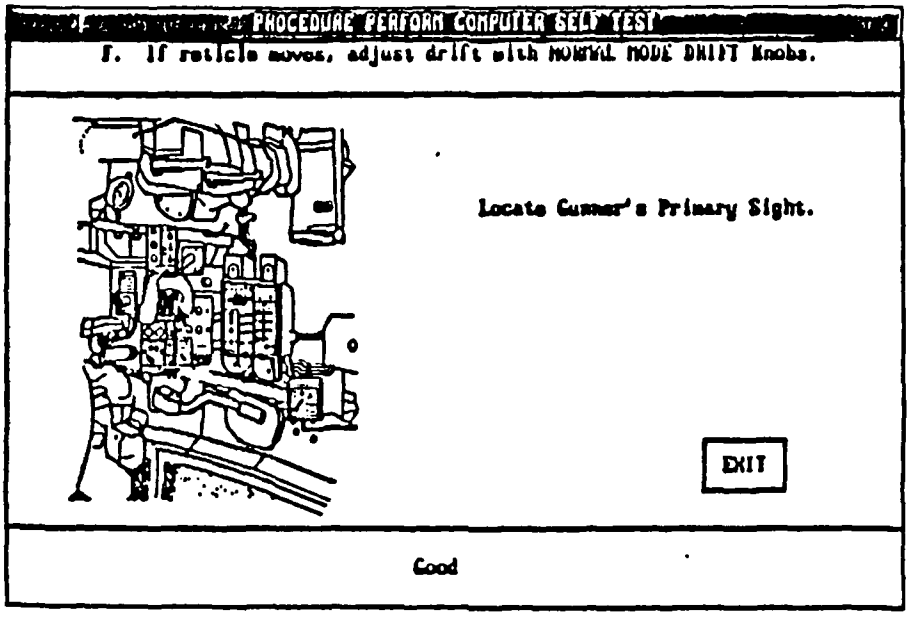


Figure 8. Sample Location Screen "Locate Gunner's Primary Sight".

- Press the right mouse button for the menu below. Select "get backdrop", and then select "gunnerov" from backdrop menu. Move the icon to the left part of screen to position picture, then press left mouse button.

  -Get Backdrop
  -Text
  -Get smart object
  -Make response area
  -Move object
  -Remove object
  -Change step pane size
  -Change message pane size
  -Exit Placement Phase
  -Exit (no saving)

- Select option "make response area". The screen will then ask if the response area is to be visible or invisible. Select visible. A prompt will appear asking you to select pen size. Press the carriage return (enter). Use the icon to mark the area on the overview to be located by the student with a rectangle.

- Select option "Exit placement phase", and then press the right mouse button for the next menu. (Note that the title at the top of the screen indicates you are in the "dependency phase" of screen preparation.) The following menu will then appear.

  -Set object condition
  -Remove prompt
  -Insert prompt
  -Invert an area
  -Uninvert an area

- Select the option "insert prompt". A message will then appear telling you that the message prompt will be removed at the conclusion of the step. You will then be instructed to type in the prompt. In this case, type "Locate gunner's primary sight". A menu will then appear asking you to select the font size. Select one of the options other than the "center options". Your icon will then take the form of a cross forming the upper left corner of a box. Move the box to the location where you want the prompt to be displayed and press the left mouse button.

- Press the right mouse button for a menu and select the option "set object condition". The menu shown below will then appear on your screen.

  - Set Conditions before step 1
  - Define step 1
  - return to dependency phase
  - view step 1 definition
  - View definition so far
  - Next step
  - step ordering
  - save screen
  - exit (no saving)

- Select option "define step 1". The screen will then ask "will student be trying to:- locate a region or locate an object. Select "locate a region". In response to the on screen prompts, you will then move the icon to the location of the gunner's site and and press the left mouse button. (Remember that you must get rid of the prompts by pressing a mouse button before you can mark the location). The message "step defined" should then appear on your screen.

- Select option "View step 1 definition". An on screen prompt will appear stating "You are now the student, use left mouse button to cause action" Next, move the icon to a position other than the correct location and press the left mouse button. The correct area should "flash", and

the message "try again" should appear in the message pane at the bottom of your screen. Now select the correct location. The message at the bottom of the screen should be "good".

- You may save the screen or simply select the exit (no save) option from the menu. In either case, you will then be returned to the menu which gives you the option of defining a new screen, moving on to another procedure or exiting the rehearser.

Creating an Action Screen

- The action screen to be created in this example is shown in Figure 9.

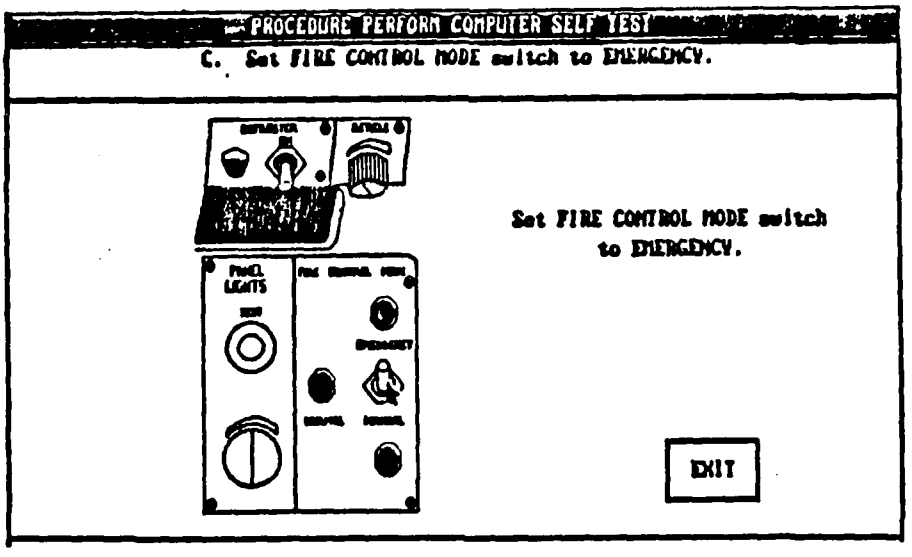


Figure 9. Sample Action Screen "Set Fire Control Mode Switch to Emergency"

- Select the "action" screen option. Press right mouse button for the placement menu and select the option "get backdrop". A menu will appear giving you the option of a color or monochrome backdrop. Select the monocrhome option.

- When the menu of backdrops appears, select the option "upperpnl".

- Select the option "get smart object". When the smart object menu appears, select the "switch" option. The next menu will allow you to select among switches varying in terms of number of positions. Select the 3-position switch. The next menu will provide a list of 3-position switches in the database. In this case, the only 3-position switch is entitled "firemode". Move the icon to "firemode" and press the left mouse button. Position the smart firemode switch next to the backdrop.

- Select option "move object". A menu will then appear with the options of moving an object by selecting it or moving an object by drawing a rectangle around it. Select the option "select object". The on screen prompt will instruct you to use your icon to indicate which smart object is to be moved and the position to which it is to be moved. After you get rid of the prompt by pressing either mouse button, the icon will take the form of a cross. Move the icon to the "smart" firemode switch and press the left mouse button, then move the icon to the switch on the backdrop and press the left mouse button. The smart switch will then disappear from your screen.

- Select the option "Exit placement phase", then press the right mouse button for the menu shown below.

  -Define dependency
  -Return to Placement Phase
  -Begin rehearsal Phase
  -Exit (no saving)

- Select the option "begin rehearsal phase". The rehearsal menu shown below will appear.

  - Set Conditions before step 1
  - Define step 1
  - return to dependency phase
  - view step 1 definition
  - View definition so far
  - Next step
  - step ordering
  - save screen
  - exit (no saving)

- Select the option "set conditions before step 1." The following menu will then appear.

  -Set object condition
  -Remove prompt
  -Insert prompt
  -Invert an area
  -Uninvert an area

- Select the option "insert prompt". A message will then appear telling you that the message prompt will be removed at the conclusion of the step.

- You will then be prompted to type in the prompt. In this case, type "Set FIRE CONTROL MODE switch to emergency". A menu will then appear asking you to select the font size. Select one of the options other than the "center options". Your icon will then take the form of a cross forming the upper left corner of a box. Move the box to the location where you want the prompt to be displayed and press the left mouse button.

- Select the option "set object condition". The message "select conditions for which smart object?" will then appear on your screen. Move the icon to the switch and press the left mouse button. A message will then instruct you to cycle through the switch positions by pressing your left mouse button and indicate the position you want (position2) by pressing the right mouse button. The message "condition defined" will then appear on your screen.

- Call up the reheaser menu again and select the option "Define Step 1". A screen message will then ask if the student will be trying to:

  -Locate a region
  -Locate an object
  -Cause Single object action
  -Allow multiple object action
  -View a note

- Select the option "cause single object action". A screen message will then instruct you to "select the object of response". Move the icon to the switch and press the left mouse button. A screen message will then instruct you to press the left mouse button to cycle through switch positions and press the right mouse button when you reach the desired position (in this case, position 1). The message "step defined" will then appear on your screen.

- Call up the rehearsal menu again, and select the option "View definition so far". The message "you are now the student. Use Left mouse button to cause action" will appear on your screen. About fifteen seconds is required for the instructions the student will see to appear in the message pane. Move the icon to the switch and use the left mouse button to cycle through switch positions until the correct position is attained, then press the right mouse button. The message "good" should then appear in the message pane.

- Call up the rehearsal menu and select the option "save screen". The system will then save the screen you have created and return you to the menu which gives you the options to define a screen (within the same procedure), create another procedure or exit the rehearser.

Creating a Note Screen

- A note screen allows students to practice using a simulated equipment component without being evaluated. In the example of a note screen previously provided (i.e., Figure 4 on page 8), a student may practice turning knobs until he/she is ready to go on to an action screen in which this skill is employed in adusting simulated equipment to certain criteria.

- Your job in preparing a note screen like that in Figure 4 differs from the preparation of an action screen in two ways. First, there is no "correct response" to be identified when preparing a note screen. Second, provisions must be made to enable students to exit a note screen and continue training the procedure.

- Select the "note screen" option and press the right mouse button for the menu shown below.

  - Get Backdrop
  - Text
  - Get Smart Object
  - Set Continue Message Location
  - Modify Continue Message
  - View Continue Message
  - Move Object
  - Remove Object
  - Change Step Pane Size
  - Change Message Pane Size
  - End Placement Phase
  - Exit (no saving)

- The sample screen to be developed includes the bare minimum requirements for a note screen (i.e., a smart object which a student can manipulate and a continuation message). This screen will contain a 3-position switch which a student can practice running through its cycles.

- Select the option "get smart object". When a menu of smart object classes appears, select the switch option. When the switch menu appears, select the three position switch option. Move the icon to the right side of the screen and press the left mouse button to place the switch on the screen.

- Select the "text" option and type the instructions in "practice cycling the switch through its positions by pressing the left mouse button".

- Select the option "set continue message location". Move the icon to the position where you would like to place the continue message and press the left mouse button. Select the option "view continue message" and the message "continue" will appear in the location you have previously selected. (Note: If you do not select a location for the continue message, the default location is in the right corner of the message screen.)

- You can modify the continue message to provide additional information for the student (e.g., "move the icon to this box and press the left mouse button when you have finished practice") by selecting the "modify continue message" option.

- At this point you will have completed preparation of the note screen, and the next step is to test the screen.

  - Select the option "exit placement phase", and you will move to the dependency phase. From the next menu, select the option "begin rehearsal phase".

  - Select the option "view definition of screen". The screen will then prompt you to respond to the instructions as if you were a student. When you finish, move the icon to the "continue" box.

  - Select the "save screen" option.

Using Screens to Address Multiple Training Objectives

- The rehearsal menus for various types of screens includes references to steps, because multiple steps may be addressed in a single screen. For example, a single screen might be used to present ten different location tasks or action tasks to a student. This is a feature which dramatically helps to reduce the cost of developing procedural simulations.

- The procedures unique to the goal of addressing multiple steps in a single screen are outlined below, by rehearser phase, in reference to a locator screen.

- During the placement phase, mark a response area for each object or region to be located.

- During the rehearsal phase, work on one "step" at a time. After you prepare the prompt and define "step 1", then select the option called "next step". Selecting this option keeps you in the rehearsal phase but the menu items change to indicate that you are working on Step 2. Prepare the prompt and definition for each step before moving on to the next step.

- When you have completed all of the steps to be addressed by the screen, select the menu option "view definition so far". The system will then instruct you to act as a student, and you will be presented with a series of location tasks.

VII. Using the Linker to Sequence Screens

Gaining Access to the Linker Menu

- Select "link screens" from the initial PROSPR menu. The next menu will provide a listing of the procedures in the database. Select the procedure which you created using the rehearser. Your screen will be cleared except for a box at the top with the word "start".

- Press the right mouse button for a menu, and the list of options shown below will appear.

  - View a Screen
  - Link Screen Icon
  - Save Linkage
  - Close Linker

Viewing Screens to be Sequenced

- The "view a screen" option allows you to check screens if you are unsure of the contents of the screen associated with a particular title. The titles used are those applied by PROSPR as you create screens (e.g., the first action screen you create is titled "action 001", the second action screen is titled "action 002").

- Select the "view a screen" option. A menu of screen titles within the procedure will appear. Move the icon to the screen of interest and press the left mouse button. In addition to the screen you have called up, a "continue" prompt will appear. When you have finished looking at the screen, move the icon to the coninue prompt and press the left mouse button.

Sequencing Screens

- Select the "link screen icon" option to begin sequencing screens. You will then be prompted to "select Icon to link FROM:". Press the left mouse button to get rid of the prompt and turn the icon from an arrow to a cross. Move the icon to the box labeled "start", and press the left mouse button. The screen will then prompt you to "select name to link TO:". Press the left mouse button to get rid of the prompt and call up the menu of screens. Note that, in addition to the screens you have created, the menu includes a "mode" and "training stratgegy" screen.

- Select the "mode" screen. The "mode" screen gives the student the option to select either an initial or advanced training session and is the first screen to be placed in any procedure. A box labeled "mode will then appear beneath the box labeled "start".

- Select the "link screen icon" option again to sequence the second screen. When you are prompted to "select icon to link FROM:", move the icon to the last box in the sequence on your screen (in this case, the box labeled "mode") and press the left mouse button. When you are prompted to "select name to link TO:", call up the menu of screens to select the screen you want to place next in the sequence (e.g., "action 001"). Notice that the option "mode" is no longer included in the menu (i.e., each time you sequence a screen, that screen is removed from the menu of screens to be linked).

- Select the second screen to be placed in the sequence and press the left mouse button. A box will then appear beneath the box labeled "mode" with the name of the screen you have just selected.

- If you should make an error while sequencing screens (e.g., selecting "action 004" when you intended to select "locator 005"), there is no way to correct the error through editing your work. Unfortunately, it is necessary to delete the entire sequence of screens for that procedure (by exiting the linker) and start the linking process over again. (NOTE: The lack of a mechanism for editing sequences of screens is a problem which needs to be addressed during future development of PROSPR).

<u>Saving Linkages, Exiting the Linker and Viewing the Procedure You Have Created</u>

- When you have finished sequencing the screens, select the "save linkage" option. To exit the "linker", select the "close linker" option. The procedure you have created will be retained in the database until you exit PROSPR. Therefore, you should check your work after exiting the linker but before exiting PROSPR.

- To view your work, select the "run a procedure" option from the PROSPR main menu. A menu will then appear listing procedures in the database. Select the procedure you have just sequenced.

IX. Summary of the Strengths and Weaknesses of PROSPR Demonstration Software

From the Perspective of Personnel Receiving Training Via PROSPR.

Over five minutes is required to load the sample procedure into memory, and this delay might need to be reduced by further development of the software. The magnitude of the delay problem is one which can be fully assessed only in the context of a larger sample of procedures. For example, if students spend five minutes calling up a procedure and five minutes training on the procedure, then fifty percent of the instructional time is "down time". If, on the other hand, students spend fifteen minutes training on the procedure, then only twenty-five percent of the instructional time is "down time". When one considers that a student might repeat the initial practice sessions many times before moving on to the advanced session, training times of thirty minutes or more (and a down time percentage of thirteen percent or less) might be realistic.

Once students are within the procedure, PROSPR is user friendly. Students can move from screen to screen with only a small delay. (Delays might increase when fully developed procedures are included within PROSPR. This possibility needs be examined in future development of PROSPR.) The actions required of students when locating equipment, assessing the status of equipment and operating equipment are clearly defined in the message pane at the bottom of each screen.

From the Perspective of Users of the PROSPR Authoring System

Overall, PROSPR provides user friendly tools which can be used to develop inexpensive procedural simulations. Most of the smart objects and the figures used for backdrops might be employed within a number of screen frames, helping to reduce the cost of preparing each screen. Further, the ability to address multiple location or action tasks with the same frame by using the "steps" option greatly reduces the cost of simulation-based training.

There are problems with the authoring system contained in the demonstration software which need to be addressed in the course of continued development of PROSPR. First, it is important to note that the authoring portion of the demonstration software contains minimal embedded training/instruction for authors. The one partial exception to this rule is found in the rehearsal editor where authors have the option of selecting

on-screen "prompts". These prompts are useful when gaining initial experience with the reheaser. When the delay caused by the appearance of these prompts becomes more of a nuissance than a help, then authors have the option of exiting the reheaser and re-entering without prompts. Unfortunately, selecting this option does not remove all of the annoying prompts.

The present menus of backdrops, smart objects and drawings use file names, and these names are limited by the MS DOS operating system to eight characters. An operational PROSPR would contain many more files than does the demonstration software, and it is doubtful whether the user could remember all of the file names. Menus with complete titles would make PROSPR much easier to use. A user friendly menu providing the full names of backdrops/objects/drawings and "pointing" to the file names, in a manner transparent to the user, would correct this problem.

The procedures for creating and "auditioning" smart objects appear to be unnecessarily time consuming. For example, after creating a smart object, one must go through a rather elaborate procedure to "audition" the object. Specifically, it is necessary to:

- save the smart object
- exit the editor for that type of smart object
- retrieve the object to be auditioned.

Several minutes are required to go through this process.

Long delays are a problem throughout the authoring process. However, the degree of severity of the problem is dependent upon how authors use PROSPR. PROSPR is designed so that an author might design, for example, a large number of "smart objects" while in the "smart object editor", creating a situation where a small percentage of an author's work session is spent getting into portions of the program. If, on the other hand, an author keeps jumping from one portion of the program to another (e.g., from backdrop editor to smart object editor) a larger percentage of each working session will be spent waiting for the system to load files into memory.